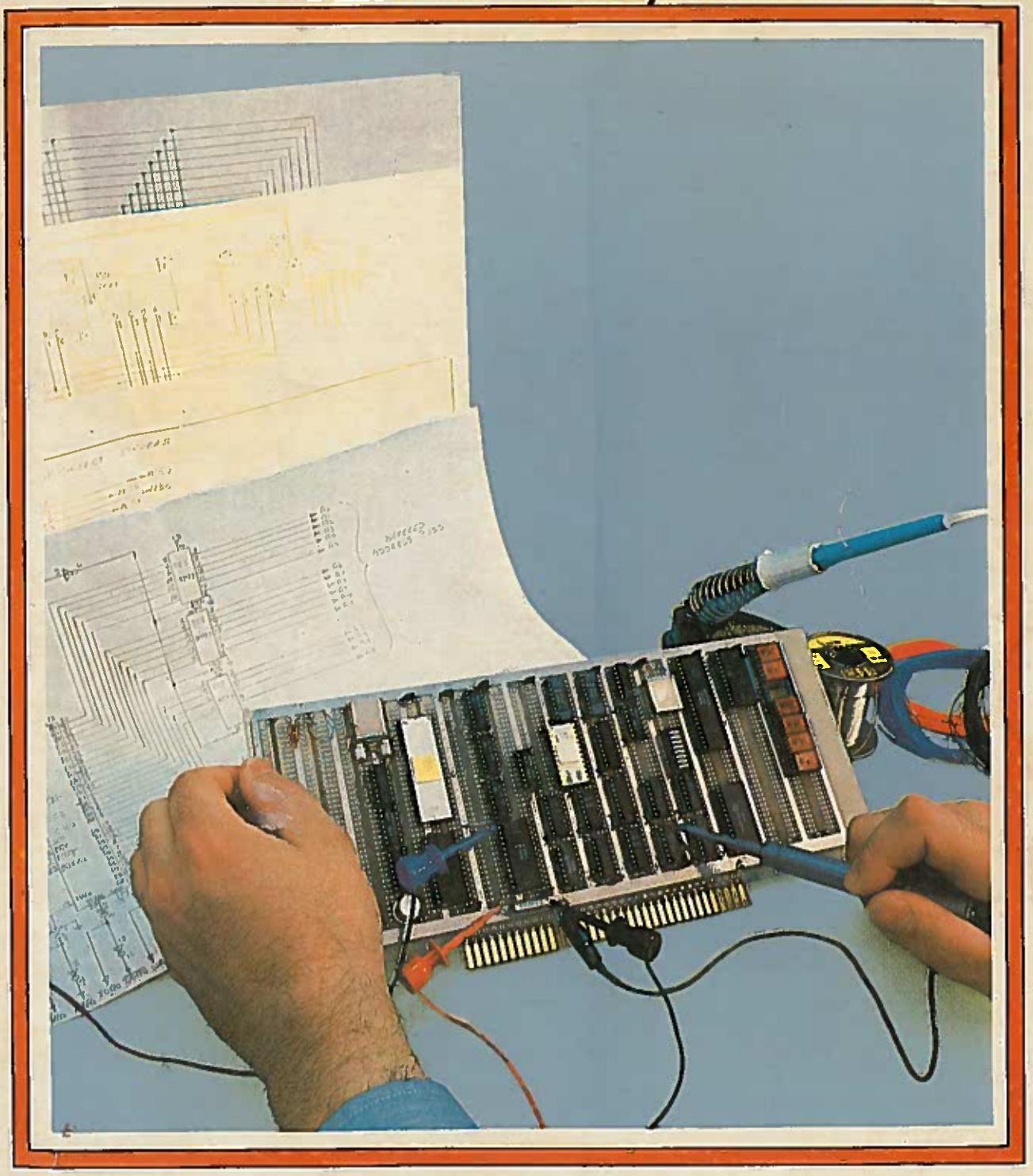
BYTE BOOKS /McGraw-Hill

Construya una MICROCOMPUTADORA basado en el Z 80

Guía de diseño y funcionamiento



Steve Ciarcia

Construya una microcomputadora basado en el Z80

(Guía de diseño y funcionamiento)

CONSULTORES EDITORIALES
AREA DE COMPUTACION E INFORMATICA

Antonio Vaquero Sánchez
Dpto. Informática y Automática
Facultad de Ciencias Físicas
Universidad Complutense de Madrid
ESPAÑA

Moisés Moshinsky A.
Universidad Autónoma Metropolitana
Unidad Azcapotzalco,
Director General de Data Steren.—México
MEXICO

Alfonso Pérez Gamma
Ingeniero Electrónico
Universidad Nacional de Colombia
COLOMBIA

José Portillo
Universidad de Lima
PERU

Construya una microcomputadora basado en el Z80

(Guía de diseño y funcionamiento)

Steve Ciarcia

Traducción:

Luis Joyanes Aguilar
Capitán de Artillería
Licenciado en Ciencias Físicas
Centro de Cálculo y Simulación Balística
Academia de Artillería de Madrid

Revisión técnica:

Antonio Vaquero Sánchez
Dpto. Informática y Automática
Facultad de Ciencias Físicas
Universidad Complutense

BYTE BOOKS/McGraw-Hill

MADRID • BOGOTA • BUENOS AIRES • GUATEMALA • LISBOA • MEXICO • NUEVA YORK
PANAMA • SAN JUAN • SANTIAGO • SAO PAULO
AUCKLAND • HAMBURGO • JOHANNESBURGO • LONDRES • MONTREAL • NUEVA DELHI
PARIS • SAN FRANCISCO • SINGAPUR • ST. LOUIS • SIDNEY • TOKIO • TORONTO

CONSTRUYA UNA MICROCOMPUTADORA BASADO EN EL Z80
(Guía de Diseño y Funcionamiento)

Prohibida la reproducción total o parcial de esta obra,
por cualquier medio, sin autorización escrita del editor.

DERECHOS RESERVADOS © 1984, respecto a la primera edición en español por
LIBROS McGRAW-HILL DE MEXICO, S.A. DE C.V.
Atlacomulco 499-501, Naucalpan de Juárez, Edo. de México
Miembro de la Cámara Nacional de la Industria Editorial, Reg. Núm. 465

ISBN: 968-451-564-2

Traducido de la primera edición en inglés de

BUILD YOUR OWN Z80 COMPUTER

Copyright © 1981, by McGraw-Hill, Inc., U.S.A.

ISBN: 0-07-010962-1

Edición exclusiva para Ediciones La Colina, S.A. (España)

ISBN: 84-85240-73-1
Depósito legal: M. 2.872-1984

Compuesto en Fernández Ciudad, S.L.
Lavel, Industria Gráfica. Humanes (Madrid)

PRINTED IN SPAIN - IMPRESO EN ESPAÑA

A mi esposa Joyce,
Steve Sunderland, y Judy y Lloyd Kishinsky

Introducción

Hace pocos años, cuando se introdujeron en el mercado los microprocesadores, el entusiasta de las computadoras y el técnico electrónico eran una misma persona. Quienes nos gustaba soldar una placa de componentes tras otra nos dormimos un poco en los laureles. Ahora, sin embargo, los precios de los sistemas completamente montados y empaquetados han caído de forma espectacular. Actualmente, cualquiera que tenga interés, casi sin necesidad de muchos conocimientos técnicos, puede adquirir y hacer funcionar una computadora. La adquisición de una nueva computadora es, ahora, similar a la compra de un aparato de televisión y se han engrosado, consecuentemente, las filas de los entusiastas de las computadoras.

Cualquier movimiento popular trae consigo un gran aumento de las publicaciones técnicas que tratan de atender la gran demanda producida. Y, en consecuencia, con la popularización de la ciencia de las computadoras, el énfasis técnico sobre la bibliografía de las computadoras se ha desplazado a partir del diseño del hardware. A diferencia con los textos introductorios denominados, por ejemplo, *«Cómo trabajan las puertas lógicas»*, la mayor parte de los libros sobre computadoras tratan el hardware de las microcomputadoras de forma simplista o intentan ser un recetario «para todo», omitiendo, a veces, algunos ingredientes muy «sabrosos». Con frecuencia, las únicas alternativas son los textos técnicos y las revistas especializadas, cuya lectura es tediosa en el mejor de los casos.

Durante varios años, he estado escribiendo una columna en la revista *BYTE* y la respuesta del lector ha demostrado que todavía existe un gran interés por el diseño del hardware y por proyectos de «hágalo usted mismo». Al mismo tiempo, me he percatado, muy a mi pesar, de que hay una carencia de materiales para dichas personas. La mayor parte de las consultas proceden de estudiantes de escuelas de segunda enseñanza, o técnicos superiores, que han leído todas las descripciones y estudiado los diagramas de bloques, pero ansian tener respuestas prácticas y ejemplos de sistemas. Lamentablemente, hay pocos libros que pueda recomendar.

Este libro, con el título de *Construya una microcomputadora basado en el Z80*, está destinado a quienes están interesados por conocer lo que hay en el interior de una microcomputadora. Está concebido para personas que, poseyendo ya un bagaje de conocimientos básicos en electrónica, desean construir una computadora por su propia cuenta en vez de adquirirlo en el comercio ya montado. No se trata de un manual de introducción a la electrónica, que comienza describiendo las puertas lógicas, ni tampoco se trata de un texto escrito solamente para estudiantes de ingeniería. Al mismo tiempo que sirve para formar al lector curioso, el objetivo de este libro es presentar un análisis práctico, paso a paso, de la arquitectura de las computadoras digitales y los detalles de una microcomputadora completa y funcional.

La computadora a construir se denomina «Z80 Applications Processor» (computadora ZAP de forma abreviada), que significa Procesador de Aplicaciones del Z80. Se basa en el microprocesador Z80 de Zilog, comercialmente disponible. Se eligió este microprocesador por su disponibilidad y bajo coste, lo mismo que ocurre con los demás componentes del ZAP. Como una ayuda adicional al entusiasta de los trabajos caseros y para aquellos experimentadores que prefieren comenzar un libro por el final, se indica en el Apéndice A un proveedor de memorias EPROM programadas. (EPROM=Memoria de sólo lectura borrable-programable.)

Se ha estructurado el libro como una secuencia lógica de «hitos» importantes en la construcción interpolados por estudios prácticos sobre la teoría de funcionamiento. El propósito del autor es doble: ayudar a un presunto constructor a

adquirir confianza en sus posibilidades y hacer la materia más asimilable mediante ejemplos concretos.

Aunque se trata secuencialmente de un manual de construcción, se presta la máxima atención a los «por qué» y a los «cómo» del diseño de la computadora. El lector dispondrá de toda clase de «herramientas»: las arquitecturas internas de los microprocesadores seleccionados, mapas de memorias, interfaces de entrada/salida, fuentes de alimentación, comunicación periférica y programación. En todos los casos, se intenta que el lector conozca el efecto de cada componente individual sobre el sistema total. Aun cuando se han documentado los detalles específicos de la computadora ZAP, se pretende que el lector sea capaz, en el futuro, de configurar una computadora de acuerdo con las necesidades del usuario. ZAP es una herramienta experimental que puede ampliarse para atender a una gran variedad de aplicaciones.

ZAP está construido como una serie de subsistemas que pueden comprobarse y hacerse funcionar con independencia. El primer elemento a construir es la fuente de alimentación. Esta es una buena forma de probar la capacidad funcional y de proporcionar un refuerzo positivo inmediato a partir de una construcción satisfactoria. La alimentación de tres tensiones está protegida contra las sobretensiones y contra las temperaturas excesivas y tiene una intensidad de corriente adecuada para un sistema ZAP ampliado.

A continuación, el lector aprenderá por qué se eligió el Z80 para el ZAP y las consideraciones estructurales que afectan a la selección de componentes en los demás subsistemas. Un capítulo completo se dedica al circuito integrado del Z80. Se explica con detalle cada señal de control y se documenta cuidadosamente cada instrucción.

La construcción del hardware prosigue en etapas con comprobación intermedia para garantizar el éxito final. Los elementos básicos de la computadora se ensamblan primero y luego se comprueban. El lector puede seleccionar qué periféricos ha de añadir. Se incluyen secciones sobre la construcción de un visualizador («display») hexadecimal, un teclado, un programador EPROM, un interfaz en serie RS-232C, un sistema de almacenamiento masivo en casete y un terminal de TRC completamente funcional. Además, un capítulo se dedica a los medios de interconexión del ZAP con las señales analógicas. Se proporcionan circuitos específicos que pueden convertir ZAP en un sintetizador digital de conversación o en un sistema de adquisición de datos y registradores de datos.

Un monitor de software especial de 1 K (1024 bytes) coordina las actividades del sistema básico de computadora y de los periféricos. El software se explica por medio de diagramas de flujo y listados con comentarios. Con este monitor como un componente integral, el ZAP puede funcionar como un terminal de computadora, un controlador dedicado o un sistema de desarrollo de software.

Este libro está concebido para los conocedores del hardware. Acorta las presentaciones teóricas sobre microcomputadoras y presenta un análisis real de las instrucciones prácticas para el lector con alguna experiencia en electrónica o para el principiante que puede acudir a alguien para la supervisión. Desde la fuente de alimentación al procesador central, este libro está concebido para quienes sienten la necesidad de entender lo que están construyendo.

Steve Ciarcia
Mayo 1981

CONTENIDO

Introducción

Capítulo 1 **Fuente de alimentación** 1

Capítulo 2 **Elementos básicos del procesador central** 21

Capítulo 3 **El microprocesador Z80** 27

Capítulo 4 **Construya su propia computadora. Comience con los elementos básicos** 95

Capítulo 5 **Los periféricos básicos** 137

Capítulo 6 **El software ZAP MONITOR (Monitor del ZAP).** 161

Capítulo 7 **Programación de una memoria EPROM** 183

Capítulo 8 **Conexión del ZAP con el mundo real** 193

Capítulo 9 **Construya un terminal de TRC** 225

Apéndice A **Técnicas de montaje** 239

Apéndice B **Códigos ASCII** 243

Apéndice C **Hojas de características técnicas según los fabricantes** 247

C1 2708 Memoria PROM borrable por UV de 8K (1K × 8). 249
C2 2716 Memoria PROM borrable por UV de 16K (2K × 8). 253
C3 2102A Memoria RAM estática de 1K × 1 bit 257
C4 2114A Memoria RAM estática de 1024 × 4 bits 261
C5 8212 Port de entrada/salida de 8 bits 265
C6 KR2376-XX Memoria ROM de codificación del teclado 273
C7 COM 2917 Transmisor/receptor asíncrono universal (UART) 277
C8 CRT 5027 Controlador y temporizador de vídeo de TRC. 285
C9 CRT 8002 Controlador de propiedades de la representación en TRC de vídeo. 295
C10 COM 8046 Generador de velocidad de transmisión en baudios. 305

Apéndice D **Sistema operativo del ZAP** 311

Apéndice E **Especificaciones técnicas de la CPU del Z80** . . . 323

E1 Especificaciones eléctricas. 325
E2 Temporización de la CPU 331
E3 Resumen del conjunto de instrucciones. 339

Glosario 343

Indice 347

Capítulo 1
FUENTE DE ALIMENTACION

No es suficiente construir una tarjeta de procesador central con algo de memoria y entrada/salida (E/S) y llamarle una computadora. Desde el momento en que examine la computadora y accione el conmutador, el sistema depende completamente del funcionamiento adecuado de su fuente de alimentación. Un libro que se refiere a la construcción de un sistema de computadora a partir de los principios básicos sería completamente inadecuado sin una descripción de cómo construir una fuente de alimentación idónea.

Mucho se ha escrito sobre el tema de las fuentes de alimentación de corriente continua. Hay convertidores de continua/continua y de alterna/continua, reguladores de shunt y conmutación, transformadores de tensiones constantes, etc. No es nuestra intención construir una fuente de alimentación fuera de lo normal, sino simplemente diseñar la fuente de alimentación de corriente continua que utilizaremos para hacer funcionar el ZAP.

En las grandes computadoras, las alimentaciones de corriente continua convierten enormes cantidades de energía para hacer funcionar miles de circuitos integrados lógicos; por necesidad, los fabricantes eligen los métodos más eficaces de conversión de energía en la alimentación. Estos métodos de la moderna tecnología serían caros y difíciles para que el aficionado los incorporara en la forma de prototipo. Afortunadamente, las exigencias de alimentación para el ZAP son muy inferiores a las que tienen las grandes computadoras; podemos sacar provecho de los métodos de diseño tradicionales e incorporar, al mismo tiempo, los más recientes adelantos en la tecnología de los reguladores. En la figura 1.1 se ilustra un diagrama de bloques de la fuente de alimentación para el ZAP.

Cada una de las tres alimentaciones de c.c necesarias para el ZAP está

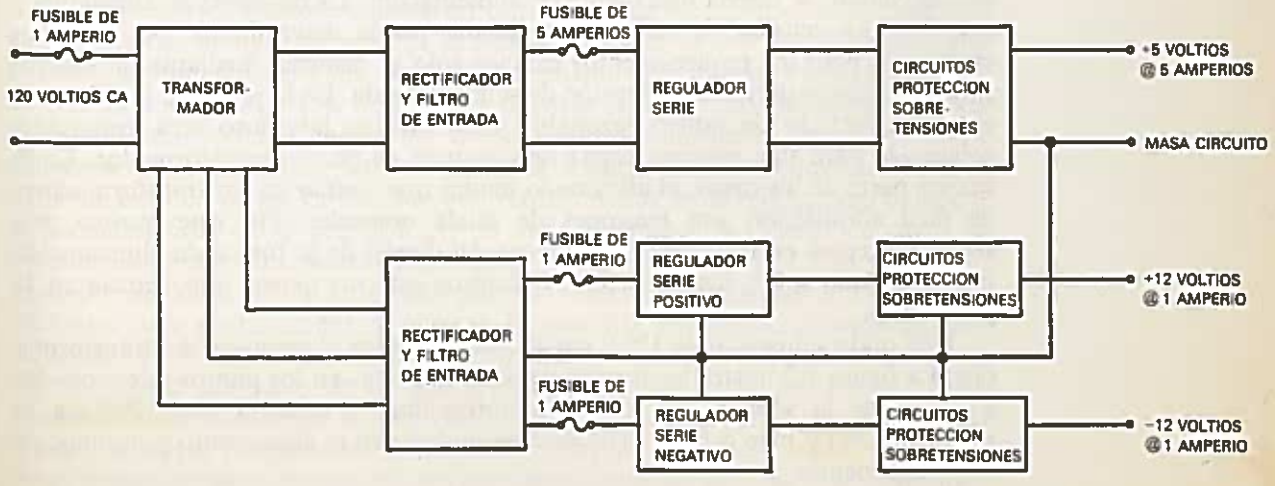


Figura 1.1 *Diagrama de bloques de la fuente de alimentación básica para el Procesador de Aplicaciones del Z80 (ZAP).*

constituida por tres módulos básicos: una sección de transformador para reducir la tensión de la red de 120 V c.a a la tensión más baja utilizada por la computadora; un rectificador-filtro de entrada para convertir c.a en c.c de pequeño rizado y un regulador que estabiliza la salida a un nivel de tensión determinado. Se examinarán, por separado, los circuitos de protección contra sobretensiones.

La especificación adecuada del transformador y filtro de entrada suele despreciarse por los aficionados, quienes hacen caso omiso de las consecuencias de un filtro deficientemente diseñado. Ello se debe, en parte, a la abundancia de información técnica puesta en circulación por los fabricantes de semiconductores que «ponen por las nubes» las virtudes de sus circuitos reguladores. A partir de esta «laguna publicitaria» se puede llegar fácilmente a la conclusión de que la sección de regulación es la única digna de consideración y, de hecho, los progresos en el diseño de reguladores y la aparición de los reguladores de tres terminales y alta potencia han disminuido la necesidad del diseñador analógico en la aplicación. Hace algunos años, unos 25 componentes discretos y bastantes cálculos eran necesarios para obtener un regulador de tensión adecuado. Ahora, sin embargo, la mayor parte de las aplicaciones pueden incluirse en un dispositivo compacto único. Aun así, una sección de filtro de entrada no debe tomarse a la ligera y requiere una consideración a fondo y una pequeña cantidad de computación para cada aplicación.

Hay tres tensiones de alimentación necesarias para hacer funcionar el ZAP. Cada alimentación lleva incorporada una sección de filtro de entrada. Debido a que la alimentación de +5 V es la más importante, recibirá la máxima atención. Para los fines de esta exposición, dividiremos la alimentación en dos secciones: transformador/filtro de entrada y regulador de salida.

En la figura 1.2 se muestra el diagrama de bloques de un filtro de entrada normalizado. En su forma más simple, está constituido por tres componentes que funcionan como sigue:

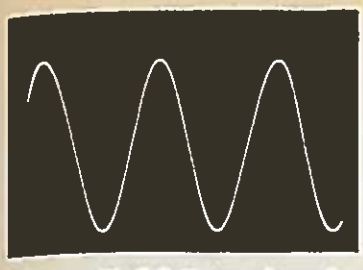
- Un transformador que aísla la alimentación de la red suministradora de energía y reduce la entrada de 120 V de c.a a baja tensión de c.a utilizable.
- Un rectificador en puente que convierte c.a en c.c de onda completa y satisface las demandas de corriente de carga del condensador del filtro.
- Un condensador de filtro que mantiene un nivel suficiente entre los ciclos de carga para satisfacer las limitaciones de las tensiones de entrada del regulador.

DISEÑO DE UN FILTRO DE ENTRADA

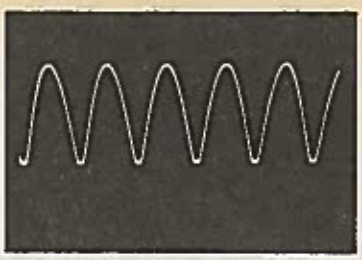
Posiblemente pensará que especificar el transformador será la primera consideración cuando se diseña una fuente de alimentación. La respuesta es afirmativa y negativa. La tensión de salida aproximada puede determinarse por métodos empíricos, pero los requerimientos exactos sólo se deducen mediante un análisis profundo que se deriva de la tensión de salida deseada. En la práctica, la diferencia entre un método de tanteo razonable y un análisis laborioso será importante solamente para una persona capaz de construir su propio transformador. En la mayor parte de los casos, el aficionado tendrá que confiar en los transformadores de fácil adquisición con tensiones de salida normales. Por este motivo, nos concentraremos en los aspectos prácticos del diseño de la fuente de alimentación más bien que sobre los pormenores técnicos que no tienen importancia en la práctica.

Una onda sinusoidal de 120 V c.a eficaces se aplica al primario del transformador. La figura 1.2 ilustra las formas de onda previstas en los puntos seleccionados a través de la sección del filtro. La fotografía 1.1 muestra que 120 V c.a es realmente 340 V pico a pico; debe tenerse cuidado en el aislamiento y montaje de los componentes.

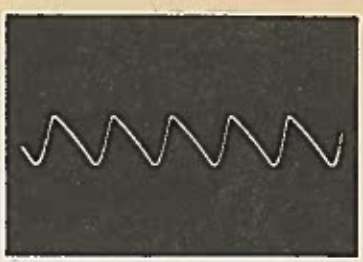
La salida del secundario del transformador será una onda sinusoidal similar, reducida en tensión. A continuación, se aplica a un puente de onda completa y la forma de onda aparecerá como en la fotografía 1.2. Observará una ligera mancha



Fotografía 1.1 *Forma de onda de entrada/salida de 120 V c.a eficaces de un transformador saturado.*



Fotografía 1.2 *Forma de onda del rectificador.*



Fotografía 1.3 *Forma de onda de rizado para diversas cargas.*

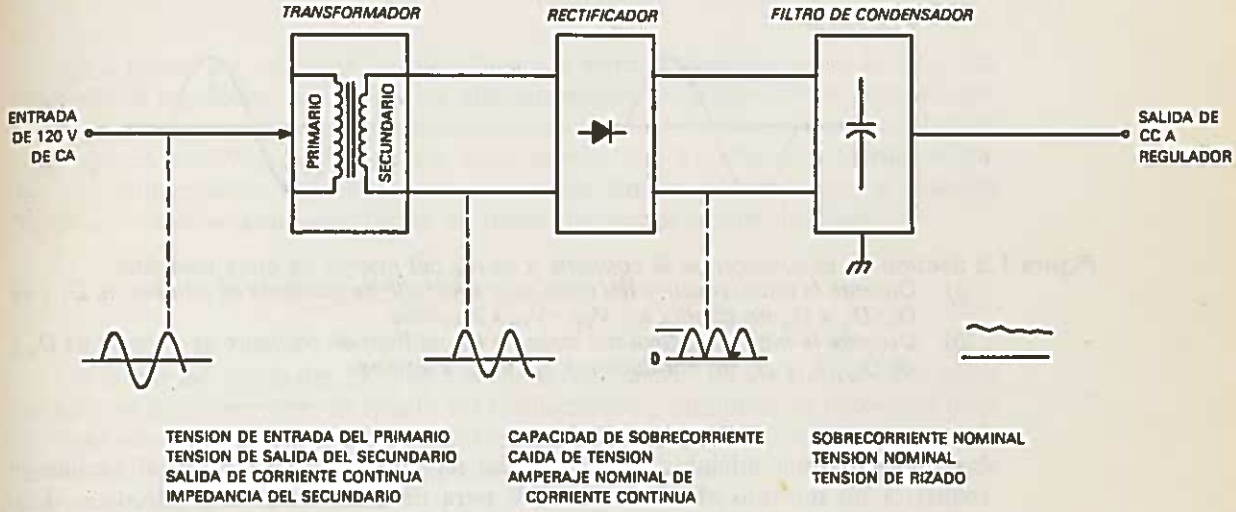


Figura 1.2 *Diagrama de bloques de un filtro de entrada estándar.*

opaca entre las «protuberancias». Como consecuencia de tratar con componentes electrónicos reales en vez de con modelos matemáticos, debemos tener conocimiento de ciertas peculiaridades. Los diodos de silicio presentan características de umbral y, de hecho, tienen una caída de tensión de 1 V a través de cada diodo. Esta caída de tensión se hace significativa en diseños de puente de onda completa y, como ilustran las figuras 1.3a, 1.3b y la fotografía 1.2, se pueden acumular cuando se añaden diodos en serie. La pérdida de 2 V en el puente es una consideración importante y debe reflejarse en los cálculos.

El regulador de tensión requiere un cierto nivel de c.c mínimo para mantener una tensión de salida constante. Si la tensión aplicada cayera por debajo de este punto, la estabilidad de salida quedará gravemente deteriorada. Por consiguiente, un condensador de salida se utiliza para suavizar las «protuberancias» en la onda sinusoidal rectificada. Cuando los diodos están conduciendo, el condensador almacena suficiente carga para mantener la tensión mínima requerida hasta el siguiente ciclo de carga. La entrada del transformador es de 60 Hz, pero debido a las características de la rectificación de onda completa, los ciclos de carga se producen a 120 Hz. El condensador se carga durante un ciclo de 8,3 ms y, como el regulador extrae potencia del mismo para satisfacer las demandas de carga, debe continuar para proporcionar, al menos, la más alta tensión de entrada mínima requerida por el regulador hasta el siguiente ciclo de carga, transcurridos 8,3 ms. Este fenómeno periódico de carga/descarga se muestra en la fotografía 1.3. La magnitud de la fluctuación de tensión entre los dos picos del ciclo se conoce como rizado. La mayor magnitud de la forma de onda, incluyendo el rizado, se designa como tensión de pico. Ambos valores son importantes de recordar y se indican en la figura 1.4.

Después de los conocimientos básicos anteriores, pasamos al caso que estudiamos concretamente: una fuente de alimentación de 5 V, 5 A. Por motivos que

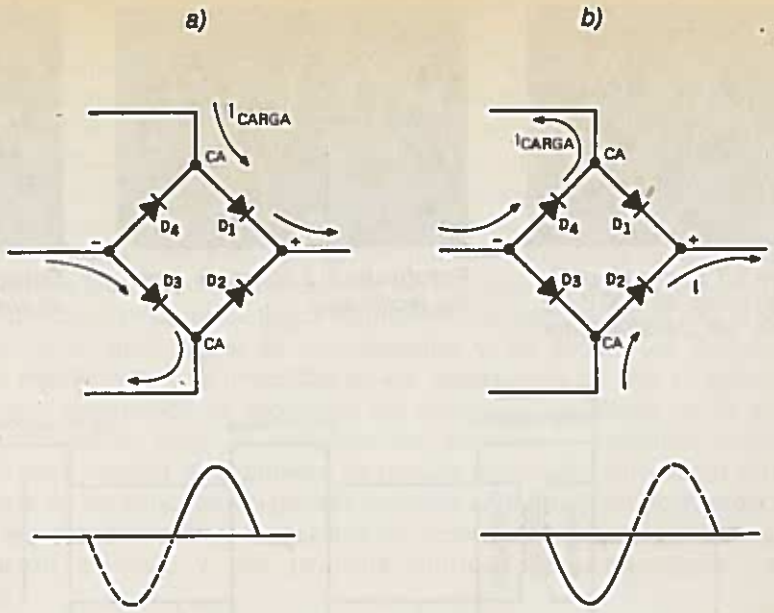


Figura 1.3 *Sentido de circulación de la corriente a través del puente de onda completa.*
a) Durante la mitad positiva del ciclo de c.a, el flujo de corriente es a través de D_1 y de D_3; D_2 y D_4 no conducen. $V_{D1}+V_{D3} \approx 2$ voltios.
b) Durante la mitad negativa del ciclo de c.a, el flujo de corriente es a través de D_2 y de D_4; D_1 y D_3 no conducen. $V_{D2}+V_{D4} \approx 2$ voltios.

expondremos más adelante, la sección del regulador de 5 V de esta alimentación requerirá un mínimo absoluto de 8,5 V para un funcionamiento adecuado. Ello significa que cualquiera que sea la magnitud de V_{PICO} y V_{RIZADO}, el nivel de V_C final no debe hacerse inferior a 8,5 V, pues de serlo el regulador no funcionará adecuadamente. Nos tomamos la libertad de que sea $V_C=10$ V y con ello facilitaremos los cálculos, sin ir en detrimento de un buen diseño. Si superáramos en mucho el valor límite de 10 V, satisfaciendo los criterios de entrada, aumentaríamos la disipación de potencia y posiblemente se destruiría el regulador. Hay una solución a este círculo vicioso y es la de *ser conservador*. La experiencia demuestra que la adición de un poco de seguridad merece la pena.

Ahora que 10 V es el objetivo, podemos seleccionar adecuadamente los otros componentes del filtro para conseguirlo. En la figura 1.5 se representa el circuito de filtro de nuestra alimentación de 5 V. R_S es la resistencia del devanado secundario del transformador. Para un transformador de 5 a 8 A, la magnitud media de R_S será 0,1 ohmios.

Los primeros valores a tener en cuenta son los siguientes:

$$V_C = V_{TENSION\ MINIMA\ DE\ ENTRADA\ DEL\ REGULADOR} = 10\ V$$
$$I_{OUT} = I_{CARGA\ REGULADOR} = 5\ A$$
$$R_S = R_{RESISTENCIA\ SECUNDARIA\ TRANSFORMADOR} = 0{,}1\ \text{ohmios}$$

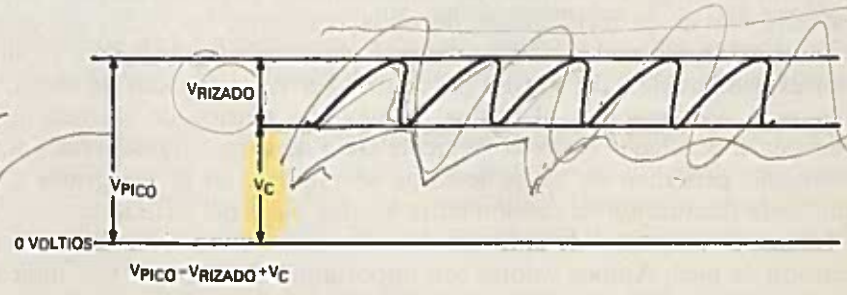


Figura 1.4 *Tensión de salida como combinación de una cierta tensión de régimen de permanente (V_C) más una tensión de rizado (V_{RIZADO}).*

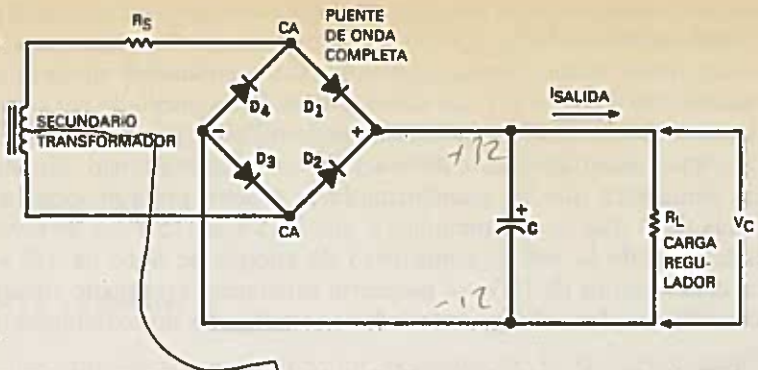


Figura 1.5 *Circuito de filtro de entrada de la fuente de alimentación de 5 V.*

V_{PICO} puede ser cualquier tensión hasta la entrada máxima para la que está diseñado el regulador. Sin embargo, ello aumentará la disipación de potencia del circuito. La regla empírica que utilizamos cuando se diseñan alimentaciones de este tipo es que V_{PICO} debe ser un 25 % mayor que V_C. De esta forma, el valor del condensador se mantendrá dentro de límites razonables. La relación $V_C/(V_{PICO}-V_C)$ se denomina factor de rizado del condensador de filtro.

$$Y_F = \frac{V_{PICO}-V_C}{V_C} = \frac{12{,}5-10}{10} = 25\,\%$$

Un factor de rizado del 25 % de 5 A caerá bien dentro de los valores nominales admisibles de la corriente de rizado del condensador y eliminará la necesidad para el diseñador aficionado de «empaparse» las especificaciones del fabricante de condensadores. El factor de rizado es arbitrario, pero es mejor mantenerlo lo más pequeño posible.

DIMENSIONAMIENTO DEL CONDENSADOR

Ahora sabemos que el condensador debe soportar una tensión de 10 V a partir de una entrada de pico de 12,5 V.

$$\left.\begin{aligned} V_{PICO} &= 12{,}5\,V \\ V_C &= 10\,V \\ V_{RIZADO} &= 2{,}5\,V \end{aligned}\right\} \quad V_C = V_{PICO} - V_{RIZADO}$$

La siguiente consideración es elegir un condensador que logre este objetivo. Otro cálculo empírico que ahorra mucho trabajo es:

$$C = \frac{dt}{dv} I$$

en donde:

C = valor del condensador en faradios = ?
I = corriente máxima del regulador = 5 A
dt = tiempo de carga del condensador = 8,3 ms (120 Hz)
dv = tensión de rizado admisible = 2,5 V

Sustituyendo los valores particulares de nuestro circuito:

$$C = \frac{(5)(8{,}3\times 10^{-3})}{(2{,}5)} = 16{,}6\times 10^{-3}\ \text{faradios}$$

o

C = 16 600 microfaradios (μF)

Los condensadores electrolíticos, que suelen estar disponibles en el comercio, tienen una tolerancia de +50 y −20 %. Tomaremos, pues, preferentemente un valor de componente normalizado de 20 000 μF. La capacidad añadida de 3400 μF reduce el rizado en otros 0,4 V y nos da algo más de «seguridad». El otro elemento a considerar con el condensador es la tensión de trabajo. Puesto que el diseño indica que V_{PICO} ha de ser 12,5 V, éste debe ser un valor satisfactorio. Sin embargo, la experiencia demuestra que los transformadores acaban por funcionar a tensiones de salida más altas que la «etiquetada» y que 12,5 V a 115 V c.a se eleva a 13,6 V cuando la tensión de la red de suministro de energía se hace de 125 V c.a. Una tensión del condensador de 15 V c.c parecería satisfacer el requisito funcional, pero es recomendable emplear el siguiente valor normalizado inmediatamente superior de 20 V c.c.

El condensador es, pues, de 20 000 μF a 20 V c.c. El rectificador puede ser un puente monolítico de onda completa o puede ser cuatro diodos discretos. Obsérvese que, puesto que un puente suele estar encapsulado, los cuatro terminales están rotulados en lugar de mostrar las marcas de polaridad de los diodos individuales. Las designaciones correspondientes a los cuatro terminales son dos terminales de entrada de c.a y dos terminales de salida + y −.

EL RECTIFICADOR

Hay tres consideraciones, cuando se elige un rectificador: sobrecargas de corriente, corriente continua y magnitud de la tensión inversa (PIV). Estas elecciones no son caprichosas y deben examinarse detenidamente.

Cuando se conecta por primera vez una fuente de alimentación, el condensador se descarga completamente. De hecho, aparecerá instantáneamente como una impedancia de 0 ohmios para la fuente de tensión. El único aspecto del circuito que limita el flujo de corriente inicial es la resistencia de los devanados secundarios del transformador y el cableado de conexión; los diseñadores suelen añadir una resistencia en serie para limitar la sobrecarga de corriente.

En este circuito, la sobrecarga es:

$$I_{SOB} = \frac{V_{PICO}}{R_S} = \frac{12,5}{0,1} = 125\,A$$

y la constante de tiempo del condensador es:

$$\tau \cong R_S \times C \cong (0,1)(20 \times 10^{-3}) = 2\,ms$$

Por regla general, la sobrecarga de corriente no producirá ningún daño al diodo si I_{SOB} es inferior a la corriente de sobrecarga del diodo y si

$$\tau < 8,3\,ms \text{ (que lo es)}$$

No podemos comprobar la magnitud nominal de la sobrecorriente hasta después de que elijamos un puente de diodos, pero pueden definirse los otros dos parámetros.

El puente puede ser uno u otro de los siguientes:

Motorola MDA 980-2: $I_{CONT} = 12\,A$, $I_{SOB} = 300\,A$, PIV = 100 V
Motorola MDA 990-2: $I_{CONT} = 27\,A$, $I_{SOB} = 300\,A$, PIV = 100 V

Los dos puentes anteriores tienen una sobrecorriente de carga de 300 A, por lo que se satisfacen también nuestras exigencias de sobrecarga.

PIV

PIV son las siglas de «peak inverse voltage» (tensión inversa de pico). Esta magnitud es la tensión máxima que puede aparecer a través del diodo antes de que

se autodestruya. Los diodos, a diferencia con los condensadores, son «implacables»; los transitorios los destruyen inexorablemente. No es extraño tener transitorios de 400 V en la línea de entrada de 115 V c.a. Ello hace que nuestra tensión nominal de 12,5 V se eleve bruscamente a 43 V (!). Por consiguiente, el rectificador de puente debe tener un valor PIV mínimo de 50 V. Por un poco más de dinero, puede conseguir un puente con un PIV de 100 V. Recuerde, la seguridad cuesta menos que las computadoras.

CORRIENTE CONTINUA

La última consideración es el valor nominal de la corriente continua. Por cuanto que el regulador puede diseñarse para un salida de 5 A, el regulador particular que hemos elegido consumirá 7 A, si se cortocircuita. No se trata de un procedimiento operativo normalizado, pero puede aplicarse en la práctica. El componente normalizado recomendado sería un puente de 50 PIV con un amperaje de 12 A. Un componente preferido sería de 100 PIV con un amperaje de 12 A o, por un aumento del coste adicional del 15%, uno de 27 A a 100 PIV. Esta última opción de diseño es estrictamente «fuerza bruta», pero salva al diodo si el condensador se cortocircuitara accidentalmente. Un transformador de 6 A podría soportar más de 12 A en condiciones de cortocircuito, pero es improbable que fuera capaz de hacer lo mismo con un amperaje de 27 A. Una u otra opción satisfará el diseño, pero solamente una salva el diseño del constructor.

EL TRANSFORMADOR

Consideremos ahora el transformador. Hemos determinado las caídas de tensión a través de los diversos componentes. Los valores se utilizan para calcular la tensión secundaria RMS (valor cuadrático medio) requerida en la forma siguiente:

$$V_{SEC(RMS)} = \frac{V_C + V_{RIZADO} + V_{RECT}}{\sqrt{2}} = \frac{10 + 2{,}5 + 2{,}0}{1{,}414} = 10{,}25\,V$$

V_{RECT} = Caída de tensión a través de cada diodo (aprox. 1 V por diodo)

En la práctica, serviría un transformador normalizado de 10 V y 6 A.

Los componentes de las alimentaciones de +12 V y de −12 V se seleccionan de una manera análoga, con la excepción de que la corriente requerida es sólo de 1 A y de que se recomienda un puente de 200 PIV debido a la configuración de rectificador particular. El esquema acabado de la sección del transformador y del filtro de nuestra computadora se ilustra en la figura 1.6.

REGULADORES DE TENSION

La siguiente consideración es la sección de regulador de tensión de nuestra fuente de alimentación. Todos los reguladores de tensión realizan la misma tarea: convierten una tensión de entrada de c.c dada en una tensión de salida de c.c estable y especifica y mantienen este punto de ajuste a través de amplias variaciones de tensión de entrada y carga de salida. El regulador de tensión típico, como se muestra en la figura 1.7, está constituido por lo siguiente:

- Un elemento de referencia que proporciona una tensión de referencia estable conocida.
- Un elemento de conversión de tensiones que muestrea el nivel de tensión de salida.

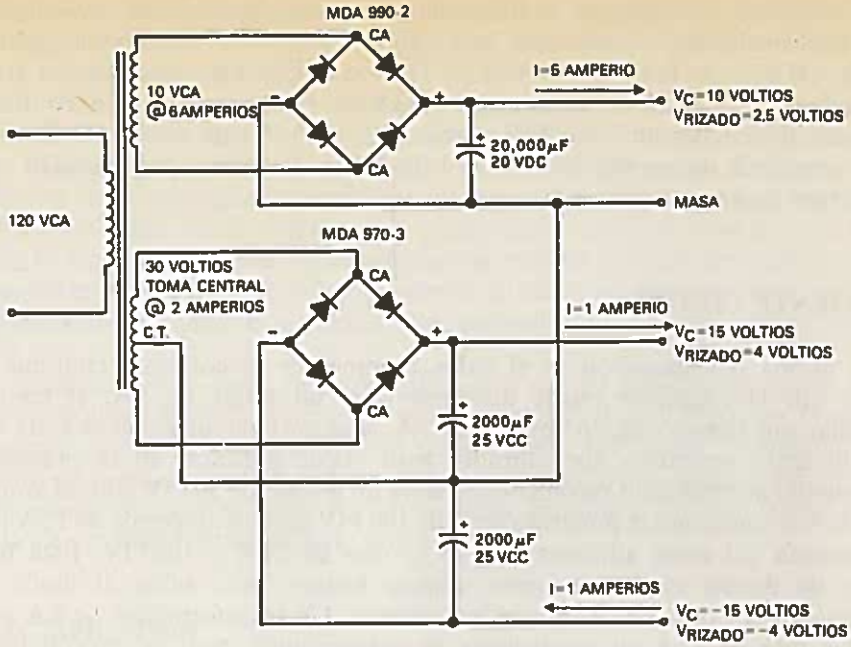


Figura 1.6 *Un diagrama esquemático de una sección de transformador y filtro de entrada.*

- Un elemento comparador que sirve para comparar el nivel de salida y el de referencia para obtener una señal de error.
- Un elemento de control que puede utilizar esta señal de error para proporcionar la conversión de la tensión de entrada para obtener la salida deseada.

El elemento de control depende del diseño del regulador y tiene una amplia variación. El control determina la clasificación del regulador de tensión: en serie, shunt o conmutación. Para el regulador en serie, el elemento de control regula la tensión de salida modulando el elemento en serie, que suele ser un transistor, y hace que actúe como una resistencia variable (figura 1.8). A medida que se incrementa la tensión de entrada, también aumenta la resistencia en serie R_S, con lo que se hace mayor la caída de tensión a través de la misma. De esta forma, la tensión de salida (V_{OUT}) se mantiene a un nivel constante.

Para realizar este control de bucle cerrado, se incorpora al hardware un sistema de comparación y realimentación de referencia. Una tensión de referencia estabilizada y fija se obtiene fácilmente por medio de un diodo Zener. Sin embargo, la

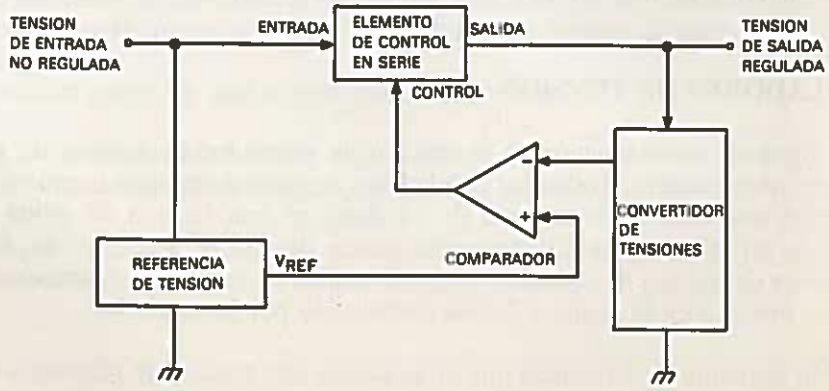


Figura 1.7 *Diagrama de bloques de un regulador de tensión típico.*

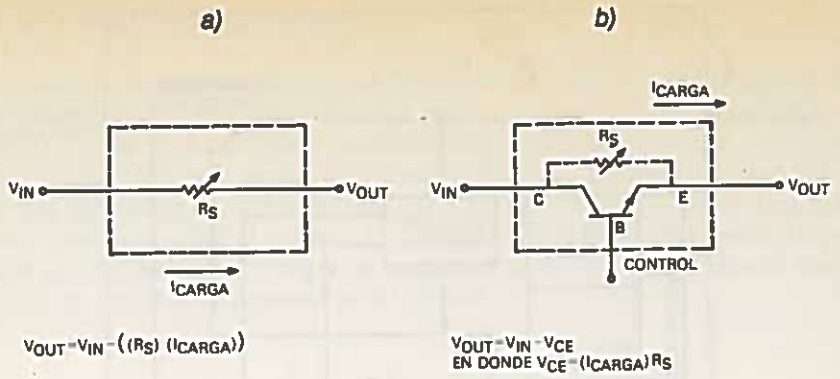


Figura 1.8 *Un elemento de control en serie en el regulador de tensión.*
a) El elemento de control en serie actúa como una resistencia variable, R_S.
b) El elemento en serie es muy frecuentemente un transistor.

corriente producida es débil; el dispositivo no podría servir como un regulador de tensión por sí mismo.

El convertidor de tensiones conectado a la salida del elemento de control en serie produce una señal de realimentación que es proporcional a la tensión de salida. En su forma más sencilla, el convertidor de tensiones es una red de divisor resistivo. Las dos señales, de referencia y de realimentación, proporcionan la información necesaria al comparador de tensiones para que se produzca la realimentación de bucle cerrado (figura 1.9). la salida del comparador excita efectivamente la base del transistor de paso en serie, de modo que la caída de tensión a través del transistor se mantendrá a un valor preajustable estabilizado cuando se reste de la tensión de entrada.

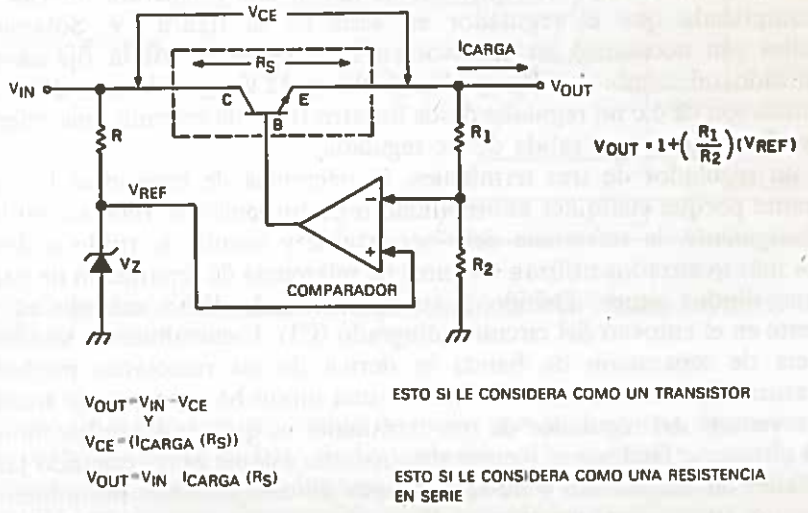


Figura 1.9 *Diagrama esquemático de un regulador de tensión en serie.*

Los diseñadores de fuentes de alimentación modernas pueden seguir utilizando componentes individuales para construir los elementos modulares de un regulador de tensión en serie, pero la mayoría reservan este esfuerzo arduo para aplicaciones especializadas. El sistema de computadora ZAP aquí descrito requiere alimentaciones de +5V, +12V y −12V. Las tolerancias combinadas de temperatura, estabilidad y de desviación no pueden superar en ± 5% cualquiera de los tres puntos de ajuste. La forma más fácil de minimizar el riesgo es reducir al mínimo el número de componentes circuitales. Otros diseñadores tenían la misma idea y, en consecuencia, se inventó el regulador de tres terminales. La figura 1.10 es el diagrama de bloques de tal dispositivo.

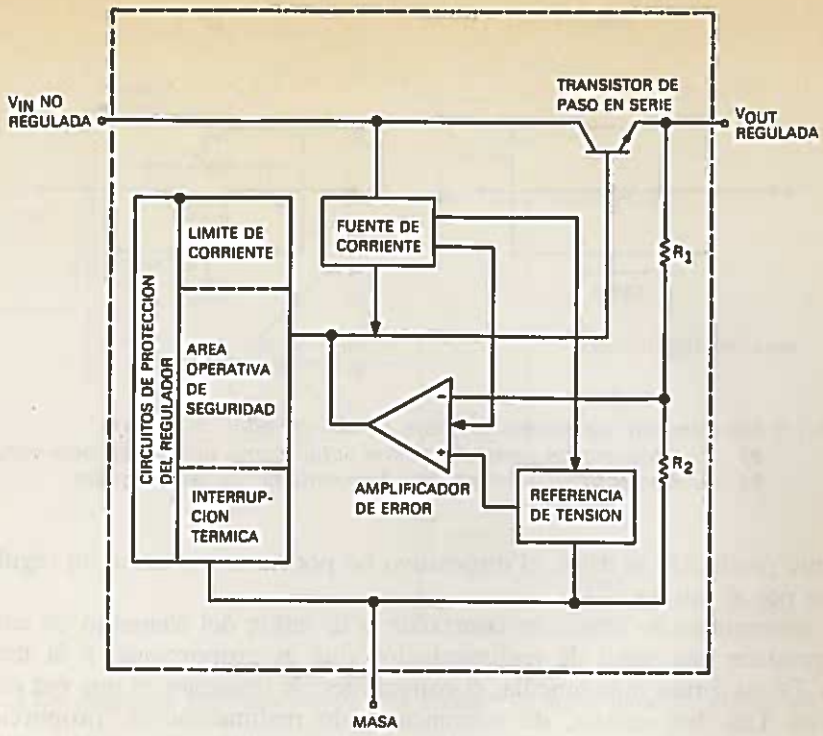


Figura 1.10 *Diagrama de bloques de un regulador de tensión de tres terminales.*

Básicamente, un regulador de tres terminales lleva incorporado todos los transistores, resistencias y diodos individuales en un solo circuito integrado. Aunque de sencillo uso, estos dispositivos tienen una estructura interna bastante más complicada que el regulador en serie de la figura 1.9. Solamente tres terminales son necesarios en aplicaciones en las que la salida fija es un valor normalizado tal como: ± 5 V, ± 6 V, ± 8 V, ± 12 V, ± 15 V o ± 24 V. Las tres conexiones son de c.c no regulada desde nuestro filtro de entrada, una referencia de masa y, finalmente, una salida de c.c regulada.

En un regulador de tres terminales, la referencia de tensión es la parte más importante porque cualquier anormalidad o perturbación se reflejará en la salida. Por consiguiente, la referencia debe ser estable y exenta de ruido o desviación. Diseños más avanzados utilizan circuitos de referencia de separación de banda más bien que diodos zener. Debido a su complejidad, dicho método es práctico solamente en el entorno del circuito integrado (CI). Esencialmente, una tensión de referencia de separación de banda se deriva de las relaciones predecibles de temperatura, de corriente y de tensión de una unión base-emisor de transistor.

Otra ventaja del regulador de tres terminales es que, en circuitos monolíticos, pueden obtenerse fácilmente fuentes de corriente estable aprovechando las buenas propiedades de adaptación y de seguimiento de componentes monolíticos. También, como en el caso anterior, el diseñador puede añadir tantos dispositivos activos como sea necesario sin aumentar significativamente la zona circuital del CI. El funcionamiento del circuito de referencia a un nivel de corriente constante reduce las fluctuaciones debidas a la variación de la tensión de la red de alimentación. De este modo, la salida ha aumentado la estabilidad. El amplificador de error se hace funcionar también a una corriente constante para reducir la influencia de la tensión de la red de alimentación.

La consideración más importante para el aficionado es que estos circuitos integrados llevan incorporados circuitos protectores, salvaguardando al regulador contra ciertos tipos de sobrecargas. Protegen al regulador contra condiciones de cortocircuitos (límite de corriente), contra la condición diferencial excesiva de entrada/salida (zona de trabajo de seguridad) y contra temperaturas excesivas de la unión (límite térmico). Por supuesto, todos estos circuitos están concebidos para proteger al regulador y no a la computadora.

ELECCION DE UN REGULADOR

El regulador de tensión híbrido μA78H05 de 5 A tiene todas las características propias del regulador monolítico de tres terminales (esto es, circuitos protectores completos). Cada cápsula TO-3 herméticamente sellada contiene un regulador monolítico μA78M05 que excita un transistor discreto de paso en serie Q1 y dos transistores detectores de cortocircuitos Q2 y Q3 (ver figura 1.11). El transistor de paso está montado sobre el mismo sustrato de óxido de berilio como la pastilla integrada del regulador, con lo que se asegura una transferencia térmica casi ideal entre Q1 y el circuito termosensible del 78M05.

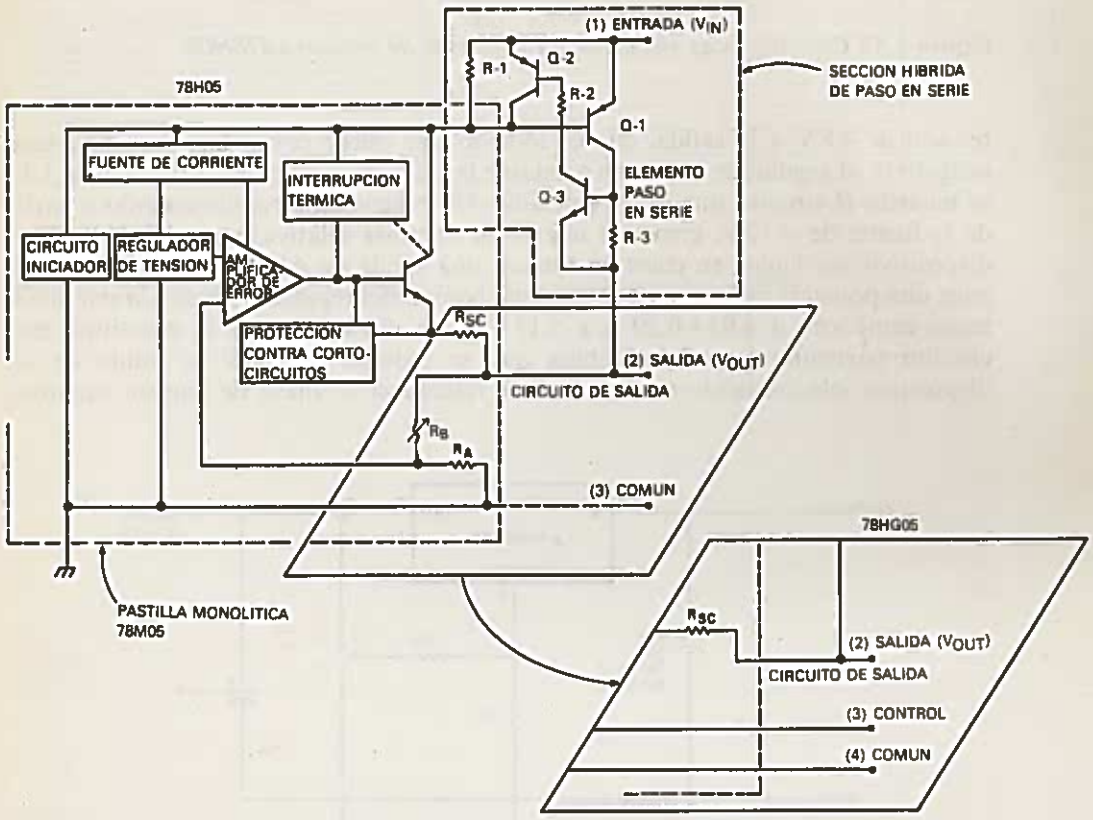


Figura 1.11 *Diagrama de bloques de un regulador de tensión híbrido μA78H05 y μA78HG05 de 5 amperios.*

El circuito de salida está concebido de modo que la exigencia de corriente del caso más desfavorable de la base de Q1, añadida a la corriente a través de R2, quede siempre por debajo del umbral de límite de corriente del 78M05. La resistencia R1, en conjunción con Q2 y Q3, constituye un circuito limitador y detector de corriente para proteger el dispositivo de paso en serie contra un consumo excesivo de corriente.

La protección de zona de seguridad se consigue por la «fuerza bruta» y está concebida pensando en el aficionado. El transistor de paso en serie es capaz de manipular la corriente de cortocircuito a la tensión máxima de entrada del 78H05. (Ver figura 1.12 para las características eléctricas del 78H05.)

La salida nominal del dispositivo es de 5,0 V, pero puede variar entre 4,8 y 5,2 V. Aun cuando caiga dentro de la tolerancia de 5,0 V $\pm$ 15 % necesaria para hacer funcionar la computadora, podría plantearse un problema con la caída de tensión en el cableado entre la fuente de alimentación y la computadora. Hasta 0,5 V podría perderse en el cableado y en los conectores. Recuérdese que a 5 A, una resistencia de sólo 0,1 ohmios puede producir una caída de 0,5 V. Lamentablemente, el 78H05 es un dispositivo de salida fija cuando se refiere a masa. Si aparece una

CARACTERISTICAS ELECTRICAS: $T_J = 25°C$, $I_{OUT} = 2.0$ A a no ser que se especifiquen de otro modo

CARACTERISTICAS	CONDICIONES	μA78H05C			UNID.
		MIN.	TIP.	MAX.	
Tensión de salida	$I_{OUT} = 2.0$ A, $V_{IN} = 10$ V	4.8	5.0	5.2	V
Regulación de línea	$V_{IN} = 8.5$ a 25 V		10	50	mV
Regulación de carga	10 mA ≤ I_{OUT} ≤ 5.0 A, $V_{IN} = 10$ V		10	50	mV
Corriente de reposo	$I_{OUT} = 0$, $V_{IN} = V_{OUT} + 5.0$ V			10	mA
Rechazo de rizado	$I_{OUT} = 1.0$ A, f = 210 Hz, 5.0 V P-P	60			dB
Ruido de salida	10 Hz ≤ f ≤ 100 kHz, $V_{IN} = V_{OUT} + 5.0$ V		40		μV_{RMS}
Tensión de vuelta al reposo (desaccionamiento)	$I_O = 5.0$ A		3.0		V
	$I_O = 3.0$ A		2.6		V
Límite corriente de cortocircuito	$V_{IN} = 10$ V		7.0		A_{pk}

Figura 1.12 *Características eléctricas del regulador de tensión μA78H05.*

tensión de 4,8 V a la salida, ello es todo lo que puede conseguir, pero podemos «engañar» al regulador haciendo ajustable la referencia de masa. En la figura 1.13 se muestra el circuito que lo hace posible. Un potenciómetro, alimentado a partir de la fuente de −12 V, crea una referencia de masa relativa para el 78H05. Si el dispositivo particular en cuestión tuviera una salida de 4,95 V y ajustáramos R1 para una potencia de 0,20 V en el terminal común del regulador, la salida referida a masa cambiaría a 4,95+0,20 o a 5,15 V. Para el fanático de la exactitud, este circuito particular permite también que se reduzca a 5,00 V la salida de un dispositivo seleccionando un terminal de referencia de masa de tensión negativa.

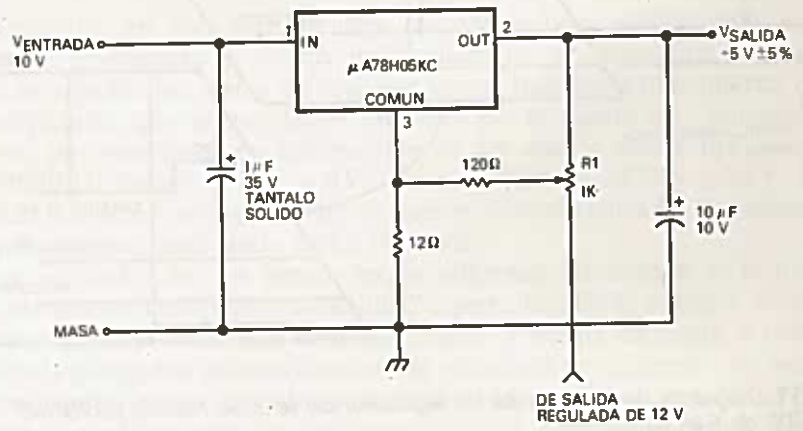


Figura 1.13 *Adición de un «ajuste fino» al regulador de tensión de tres terminales μA78H05.*

Con la alimentación de 5 V completa, nuestro próximo objetivo lo constituyen las alimentaciones de +12 V y de −12 V. Otros dispositivos dentro de la familia de reguladores 7800 satisfarán los requisitos. Los reguladores 7812 y 7912 son reguladores de 1 A, positivo y negativo, respectivamente; presentan las mismas características de protección que el 78H05. En las figuras 1.14 y 1.15 se indican las especificaciones exactas. Puesto que estamos tratando de corrientes mucho más bajas que la alimentación de +5 V, hay considerablemente menos interés por las pérdidas de tensión a través de los cables de conexión y resulta innecesario añadir circuitos de ajuste fino. En la figura 1.16 se ilustra el esquema final de la fuente de alimentación del ZAP. Se añaden los esquemas de reguladores adicionales (figuras 1.17a, b, c y d) para demostrar cómo la serie 7800 de los reguladores puede utilizarse en nuestra aplicación. ¿Ya hemos acabado? Por supuesto, no. Un examen detenido de la figura 1.16 indica dos elementos no examinados anteriormente: radiadores de calor y protección contra sobretensiones. Estos dos temas y un breve estudio de la importancia de la disposición correcta completarán el capítulo.

μA7812

CARACTERISTICAS ELECTRICAS: V_{IN} = 19 V, I_{OUT} = 500 mA, −55° C ≤ T_J ≤ 150° C, C_{IN} = 0,33 μF, C_{OUT} = 0,1 μF, a no ser que se especifiquen de otro modo.

CARACTERISTICAS		CONDICIONES		MIN.	TIP.	MAX	UNID.
Tensión de salida		T_J = 25° C		11,5	12,0	12,5	V
Regulación de línea		T_J = 25° C	14,5 V ≤ V_{IN} ≤ 30 V		10	120	mV
			16 V ≤ V_{IN} ≤ 22 V		3,0	60	mV
Regulación de la carga		T_J = 25° C	5 mA ≤ I_{OUT} ≤ 1,5 A		12	120	mV
			250 mA ≤ I_{OUT} ≤ 750 mA		4,0	60	mV
Tensión de salida		15,5 V ≤ V_{IN} ≤ 27 V 5 mA ≤ I_{OUT} ≤ 1,0 A P ≤ 15 W		11,4		12,6	V
Corriente de reposo		T_J = 25° C			4,3	6,0	mA
Cambio de corriente de reposo	con línea	15 V ≤ V_{IN} ≤ 30 V				0,8	mA
	con carga	5 mA ≤ I_{OUT} ≤ 1,0 A				0,5	mA
Tensión de ruido de salida		T_A = 25° C, 10 Hz ≤ f ≤ 100 kHz			8	40	μV/V_{OUT}
Rechazo de rizado		f = 120 Hz, 15 V ≤ V_{IN} ≤ 25 V		61	71		dB
Tensión de desaccionamiento (vuelta al reposo)		I_{OUT} = 1,0 A, T_J = 25° C			2,0	2,5	V
Resistencia de salida		f = 1 kHz			18		mΩ
Corriente de cortocircuito		T_J = 25° C, V_{IN} = 35 V			0,75	1,2	A
Corriente máxima de salida		T_J = 25° C		1,3	2,2	3,3	A
Coeficiente de temperatura media de tensión de salida		I_{OUT} = 5 mA	−55° C ≤ T_J ≤ +25° C			0,4	mV/° C/ V_{OUT}
			+25° C ≤ T_J ≤ +150° C			0,3	

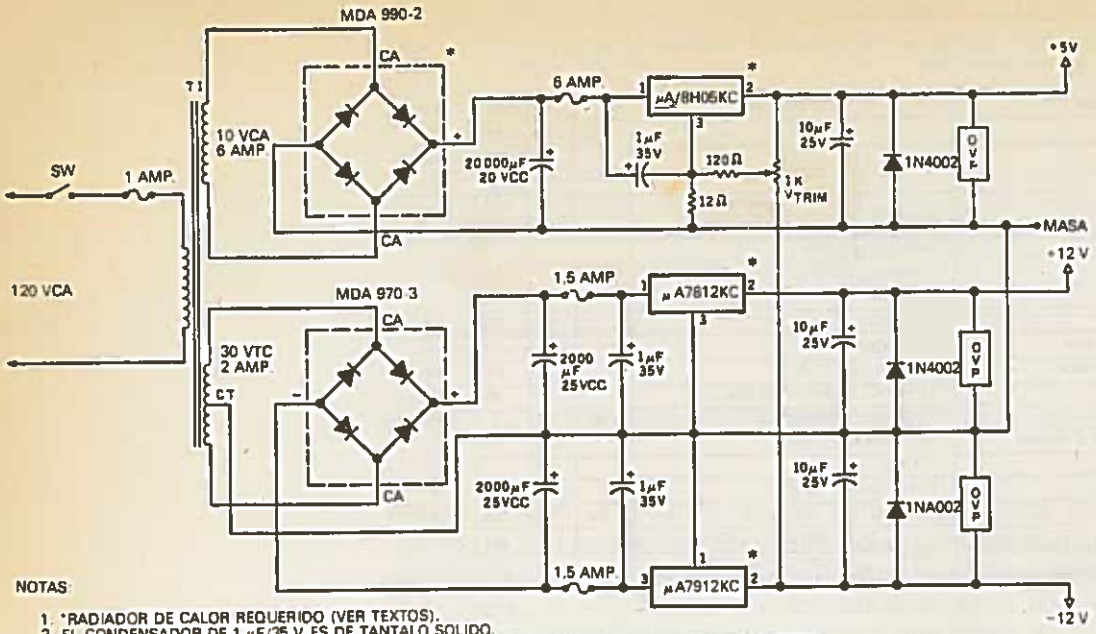
Figura 1.14 *Características eléctricas del regulador de tensión μA7812.*

μA7912

CARACTERISTICAS ELECTRICAS: V_{IN} = −19 V, I_{OUT} = 500 mA, C_{IN} = 2 μF, C_{OUT} = 1 μF, −55° C ≤ T_J ≤ 150° C, a no ser que se indique de otro modo.

CARACTERISTICAS		CONDICIONES		MIN.	TIP.	MAX.	UNID.
Tensión de salida		T_J = 25° C		−11,5	−12,0	−12,5	V
Regulación de línea		T_J = 25° C	−14,5 V ≤ V_{IN} ≤ −30 V		10	120	mV
			−16 V ≤ V_{IN} ≤ −22 V		3,0	60	mV
Regulación de la carga		T_J = 25° C	5 mA ≤ I_{OUT} ≤ 1,5 A		12	120	mV
			250 mA ≤ I_{OUT} ≤ 750 mA		4,0	60	mV
Tensión de salida		−15,5 V ≤ V_{IN} ≤ −27 V 5 mA ≤ I_{OUT} ≤ 1,0 A p ≤ 15 W		−11,4		−12,6	V
Corriente de reposo		T_J = 25° C			1,5	3,0	mA
Cambio de corriente de reposo	con línea	−15 V ≤ V_{IN} ≤ −30 V				1,0	mA
	con carga	5 mA ≤ I_{OUT} ≤ 1,0 A				0,5	mA
Tensión de ruido de salida		T_A = 25° C, 10 Hz ≤ f ≤ 100 kHz			25	80	μV/V_{OUT}
Rechazo de rizado		f = 120 Hz, −15 V ≤ V_{IN} ≤ −25 V		54	60		dB
Tensión de desaccionamiento (vuelta al reposo)		I_{OUT} = 1,0 A, T_J = 25° C			1,1	2,3	V
Corriente máxima de salida		T_J = 25° C		1,3	2,1	3,3	A
Coeficiente de temperatura media de tensión salida		I_{OUT} = 5 mA, −55° C ≤ T_J ≤ 150° C				0,3	mV/° C/ V_{OUT}
Corriente de cortocircuito		V_{IN} = −35 V, T_J = 25° C				1,2	A

Figura 1.15 *Características eléctricas del regulador de tensión μA7912.*



NOTAS:

1. *RADIADOR DE CALOR REQUERIDO (VER TEXTOS).
2. EL CONDENSADOR DE 1 μF/35 V ES DE TANTALO SOLIDO.
3. OBSERVESE QUE HAY UNA DIFERENCIA DE DESIGNACION DE TERMINALES ENTRE EL 7812 Y EL 7912.
4. EL FUSIBLE ESTA UNIDO A LA ENTRADA DEL REGULADOR Y ENTRE EL CONDESADOR DE FILTRO Y EL PUENTE DE DIODOS.

Figura 1.16 *Esquema de la fuente de alimentación acabada para la computadora ZAP.*

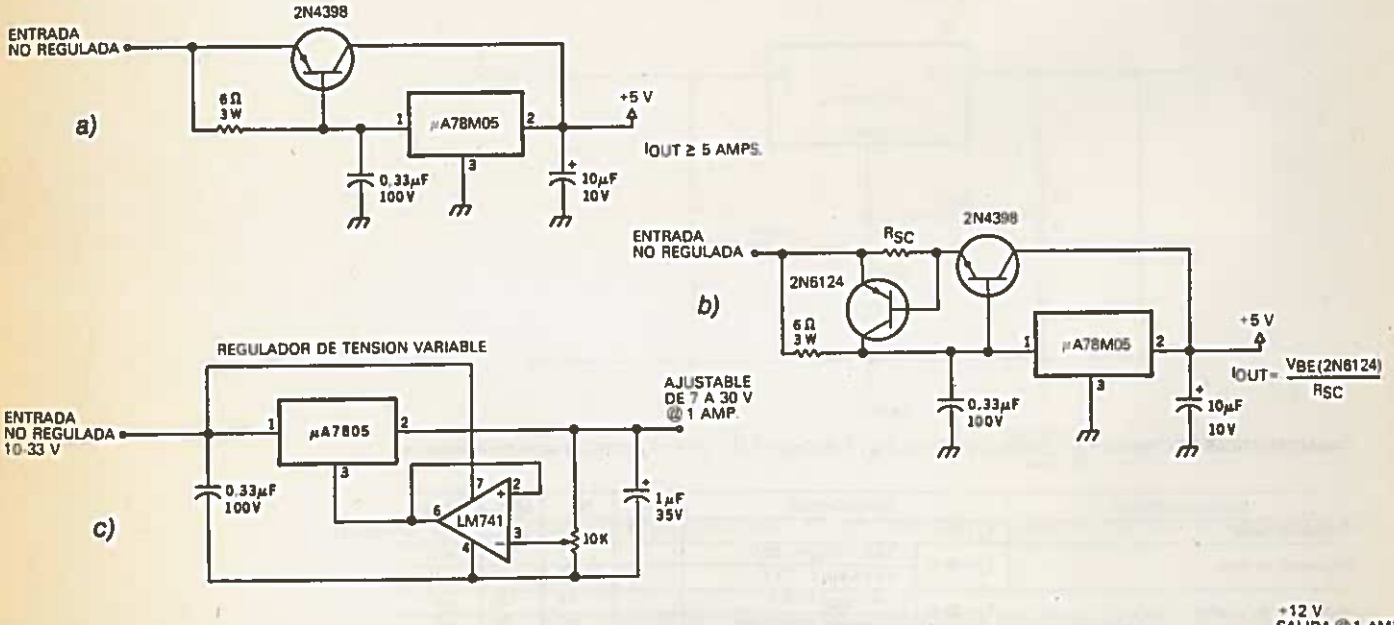


Figura 1.17 *Esquemas reguladores de tensión para demostrar cómo puede utilizarse la serie 7800 de reguladores.*

- *a) Un regulador de tensión de alta corriente con el empleo de un regulador de tres terminales 78M05 de 500 mA.*
- *b) Un regulador de tensión de alta corriente, protegido contra corto-circuitos, en una versión ampliada de la figura 1.17a.*
- *c) Con el empleo de un regulador de tensión 7805+5 V para obtener una tensión de salida más alta.*
- *d) Un regulador doble de tensión de seguimiento de ±12 V.*

LA DISPOSICION GENERAL ES IMPORTANTE

En los reguladores de circuitos integrados se emplean transistores de banda ancha en su construcción para optimizar la respuesta. En consecuencia, deben compensarse adecuadamente para asegurar un funcionamiento estable de bucle cerrado. Su compensación puede perturbarse por la capacidad parásita y la inductancia de línea de una disposición general inadecuada. Las longitudes de los cables del circuito deben mantenerse en un mínimo y los condensadores de paso o desvío externos, en particular, deben situarse tan cerca como sea posible del circuito de control del regulador.

En la figura 1.18a se ilustra una disposición general típica de los componentes de nuestra alimentación y en la figura 1.18b se detalla las zonas que pueden plantear problemas. Una colocación inadecuada del condensador de entrada puede dar lugar a un rizado indeseable en la tensión de salida. Ello se produce cuando la corriente que circula por el circuito de entrada influye en la línea de masa común del regulador. La caída de tensión producida a través de R2' hará que la salida del regulador fluctúe de la misma manera que el circuito de ajuste fino de la tensión anteriormente estudiado. Las corrientes máximas en el circuito de entrada (que está constituido por el rectificador y el condensador de filtro) pueden ser del orden de magnitud de decenas de amperios durante ciclos de carga. Estas sobrecorrientes pueden dar lugar a importantes caídas de tensión en los cables largos o conexiones de hilos delgados. También pueden degradar el rendimiento hasta el punto de que la adecuada tensión de entrada al regulador no puede mantenerse salvo durante el funcionamiento a baja corriente.

El bucle de corriente de salida es también susceptible a la disposición general del circuito. En un regulador de tres terminales, la tensión de salida fija $V_{OUT(REG)}$ se refiere entre los terminales «out» y «común» de la pastilla integrada del circuito. Debido a la corriente de carga que circula a través de R2', R3' y R4', así como a la propia carga, estas pérdidas de tensión combinadas pueden reducir V_{OUT} a un nivel inadmisible. Obsérvese que la masa para este circuito está en el punto C, mientras que la carga R presente está entre los puntos A y B. Si otra carga, más memoria por ejemplo, se conecta a esta alimentación entre los puntos A y C, tendría una V_{OUT} diferente. El ajuste fino de dicha alimentación basculante, con inversión de

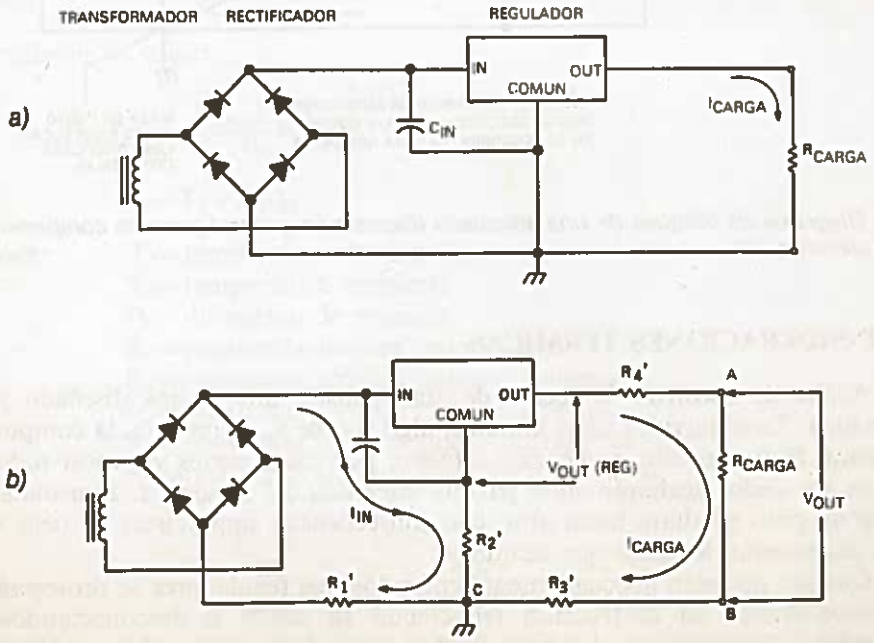


Figura 1.18 *Una disposición general típica de los componentes de la fuente de alimentación y problemas asociados.*

a) Una disposición general típica.

b) Errores aportados por la disposición general de la figura 1.18a.

polaridad por realimentación negativa, puede ser peligroso; es posible tener una carga completamente dentro de la tolerancia y otra con tensión excesiva o deficiente. Un último punto a considerar es que R4' sirve para anular la finalidad del regulador porque reduce continuamente V_{OUT} a medida que aumenta la corriente de carga.

En la figura 1.19 se ilustra el diagrama de bloques de una disposición general adecuada. Todos los recorridos de alta corriente deben utilizar hilo grueso para reducir al mínimo la resistencia y las caídas de tensión resultantes. Observará que los recorridos de los circuitos de entrada y de salida están efectivamente separados. Ha de constatarse que los hilos de conexiones procedentes del rectificador van directamente al condensador y que los hilos procedentes del condensador envían energía al resto del circuito. Si sigue este convenio y emplea dos pares de hilos conductores separados, puede eliminar los errores inducidos por el circuito de entrada.

Finalmente, necesitamos examinar el concepto de masa de punto único. Un solo punto en la fuente de alimentación debe designarse como masa; las masas de todas las demás alimentaciones y cargas están conectadas a dicho punto. En la práctica, la mejor forma de poner en práctica esta conexión de masa es utilizar una banda metálica o varios trozos de hilo grueso soldados juntos. La banda es un conductor colectivo (bus) de masa con una resistencia tan baja que una tensión medida entre el punto A y cualquier lugar a lo largo del bus será virtualmente no detectable. Otro bus de +5 V debe conectarse a la salida de la alimentación de modo que sea uniforme la distribución de tensión a través de todo el circuito. Ha de emplearse hilo grueso en las fuentes de alimentación. Aunque el hilo de «resistencia cero» no es fácilmente conseguible, recuérdese siempre que lo mejor es que sea el hilo lo más grueso posible.

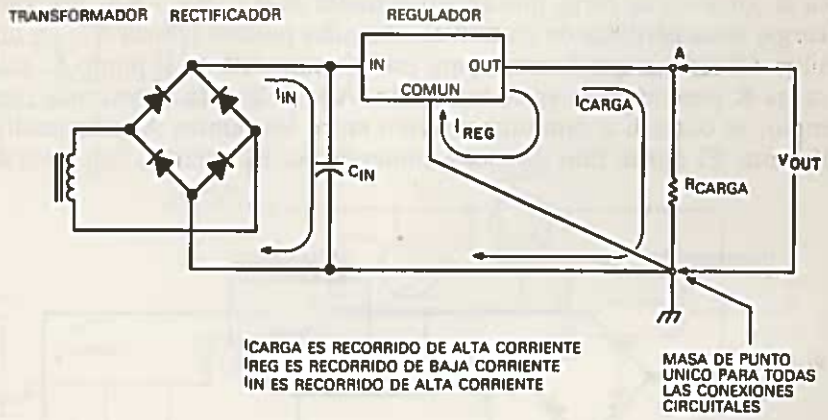


Figura 1.19 *Diagrama de bloques de una adecuada disposición general para los componentes de la fuente de alimentación.*

CONSIDERACIONES TERMICAS

Acaba de construir la fuente de alimentación que hemos diseñado y todo funciona. Transcurridos unos minutos, algo sucede y, de repente, la computadora se para. Naturalmente, comenzará a buscar por todas partes y a tocar todo. A la larga, sus dedos acabarán en la pastilla integrada del regulador. Inmediatamente dará un grito y saltará hacia atrás con consecuencias imprevistas. Si tiene suerte, sus dedos serán lo único que se queme.

Cuando no estén adecuadamente enfriados, los reguladores se protegerán a sí mismos contra su destrucción reduciendo su salida o desconectándose por completo. En este caso, el sistema podría dejar de funcionar. Un problema más catastrófico se plantea con los circuitos integrados que emplean tres tensiones para el funcionamiento normal. La pérdida de una o más de estas tensiones podría deteriorar permanentemente el dispositivo. Ello nunca sucederá si está limitada la disipación de potencia y se emplean adecuados métodos de enfriamiento.

El primer paso es comprobar la disipación de potencia de nuestro diseño con los valores nominales de los dispositivos particulares. En términos prácticos, la potencia, expresada en vatios, es el producto de la tensión (en voltios) por la intensidad de corriente (en amperios):

$$P_D = E \times I$$

En nuestro regulador de 5 V, tenemos $V_C = 10\,V$ y $V_{PICO} = 12{,}5\,V$ a 5 A.

$$P_{D(NOM)} = (V_C - V_{OUT}) \times 5\,A =$$
$$= 10 - 5) \times 5 = 25\,W$$
$$P_{D(PICO)} = (V_{PICO} - V_{OUT}) \times 5\,A =$$
$$= (12{,}5 - 5) \times 5 = 37{,}5\,W$$
$$P_{D(MEDIA)} = \frac{37{,}5 + 25}{2} = 31{,}25\,W$$

Ello significa que en condiciones de plena carga, se producirá unos 30 W de calor por el 78H05. El dispositivo está afortunadamente calculado para 50 W a 25 °C y todavía es capaz de trabajar 30 W hasta 75 °C.

Aunque esté limitada la disipación de potencia interna, la temperatura de la unión debe mantenerse por debajo de la temperatura máxima especificada (125 °C) para que el dispositivo funcione adecuadamente. Para calcular el radiador de calor requerido, hay ecuaciones específicas que resolver.

Los datos térmicos y los cálculos requeridos son los siguientes:

θ_{JC} típico = 2,0 $\qquad$ θ_{JC} máx = 2,5
θ_{JA} típico = 32 $\qquad$ θ_{JA} máx = 38

$$P_{D(MAX)} = \frac{T_{J(MAX)} - T_A}{\theta_{JC} + \theta_{CA}} \qquad \text{para } \theta_{CA} = \theta_{CS} + \theta_{SA}$$

Resolviendo para T_J,

$$T_J = T_A + P_D(\theta_{JC} + \theta_{CA})$$

o sin radiador de calor:

$$P_{D(MAX)} = \frac{T_{J(MAX)} - T_A}{\theta_{JA}}$$

$$T_J = T_A + P_D \theta_{JA}$$

en donde:
T_J = temperatura de la unión
T_A = temperatura ambiente
P_D = disipación de potencia
θ_{JC} = resistencia térmica unión a caja
θ_{JA} = resistencia térmica unión a ambiente
θ_{CA} = resistencia térmica caja a ambiente
θ_{CS} = resistencia térmica caja a radiador de calor
θ_{SA} = resistencia térmica radiador a ambiente

$$\theta_{JA} = \frac{T_J - T_A}{P_D} = \frac{125°C - 25°C}{31{,}25\,W} = 3{,}2\,°C/W$$

Puesto que θ_{JA} en su valor calculado es inferior al de θ_{JA} que figura en las hojas de características, se requiere definitivamente un radiador de calor y uno del tipo TO-3 de 3,2 °C/W es el mínimo deseado.

Antes de que dimensione un radiador para el 78H05, ha de percatarse de que hay dos reguladores más y dos rectificadores de puente que necesitarán radiador.

Cada regulador de 12 V tendrá una disipación media de 5 W. El puente de diodos asociado con la alimentación de +5 V (recuérdese la caída de 2 V) disipa unos 10 W, mientras que el otro proporciona 2 W. Por consiguiente, cualquier radiador en la fuente de alimentación debe soportar más de 50 W.

¿CUAL ES EL METODO PRACTICO PARA ELEGIR LOS RADIADORES DE CALOR?

La elección de un radiador de calor puede ser fácil, o difícil, dependiendo de sus aptitudes para las medidas empíricas. Ya sabemos que necesitamos un radiador de 50 W. Es fácil suponer que comprando «uno calculado para 50 W» de un proveedor electrónico local se resolverá el problema. Lo que ello suele significar, sin embargo, es que si se aplica 50 W, a través de un transistor, a este radiador y la temperatura ambiente es 25 °C, la temperatura superficial del radiador se elevará a 100 °C (¡temperatura apta para freír huevos!).

No hemos de olvidar que las especificaciones del fabricante siempre están concebidas para limitar la temperatura máxima de la unión, y no a mantener la caja suficientemente fría para que se pueda tocar. Personalmente, el autor detesta las fuentes de alimentación calentadas al rojo. El conseguir un radiador que admitiera nuestros 50 W y trabajara permanentemente con una temperatura de 60 a 70 °C probablemente significaría obtener un radiador calculado para 200-300 W y no hay que olvidar que los radiadores son caros... y grandes.

La solución más sencilla es la mejor. Preferimos el enfriamiento por aire a presión. Ponga los 50 W en un radiador económico de, por ejemplo, 100 W de valor nominal y gaste su dinero en un buen ventilador. También puede repasar todos los cálculos y determinar cuántas pulgadas cuadradas necesita, pero el efecto de soplar un poco de aire sobre un radiador amplía enormemente sus posibilidades.

PROTECCION CONTRA SOBRETENSIONES

El área final a tratar en la fuente de alimentación es la protección contra sobretensiones. Tal como se diseñan por los fabricantes, los reguladores se protegen, por sí mismos, reduciendo la tensión de salida o desconectándose por completo. Las posibilidades de deterioro de los componentes de la computadora por baja tensión son muy pequeñas en comparación con los daños por sobretensiones. Es improbable que suceda, pero si se cortocircuitara accidentalmente el 78H05, una tensión tan alta como de 12,5 V se aplicaría al bus de +5 V y, entonces, se podría «despedir» de la computadora.

Protección contra sobretensiones de 5 V

D_1	5,6 V	1N4734
SCR_1	50 V 25 A	2N682
Fusible	6 amp	fusión rápida

Protección contra sobretensiones de 12 V

D_1	13 V	1N4743
SCR_1	50 V 8 A	2N4441
Fusible	1,5 amp	fusión rápida

Los componentes semiconductores de esta protección contra sobretensiones de 12 V se invierten en polaridad para la de −12 V.

El circuito de la figura 1.20 es un protector contra sobretensiones (OVP) simple. Puede utilizarse, como se indica, con las alimentaciones de 5 y 12 V. Los componentes adecuados están indicados en las tablas de la figura 1.20. Observará que los fusibles están calculados para valores más altos que los que anteriormente examinamos. El fusible es para el OVP y no para proteger a los reguladores. Lamentablemente, la naturaleza de los fusibles del tipo de fusión rápida no es la de dejar pasar 5 A (si es un fusible de 5 A). El fusible debe tener un amperaje más alto para permitir un funcionamiento del circuito a 5 A. Puesto que la corriente de

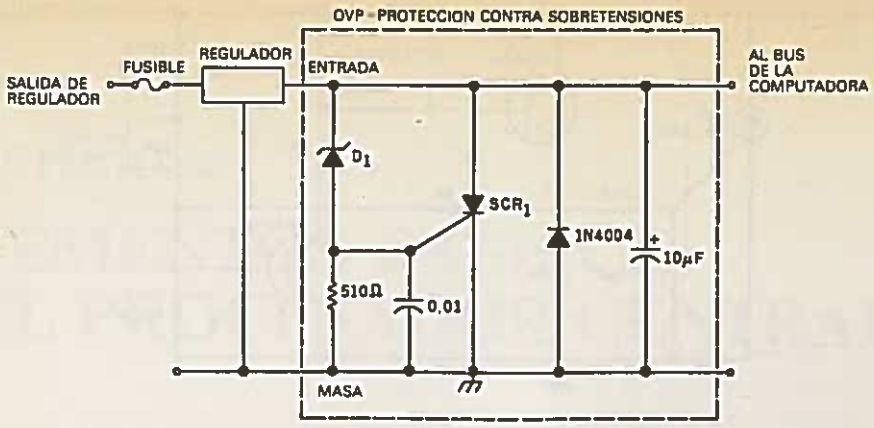


Figura 1.20 *Un circuito simple de protección contra sobretensiones.*

cortocircuito del 78H05 es de 7 A, el rectificador controlado de silicio (SCR) de 25 A seguramente hará más corta la vida útil del fusible si se dispara. En las figuras 1.21 y 1.22 se ilustran circuitos de OVP algo más complejos y también pueden utilizarse.

¿Qué hace un OVP (que suele denominarse «cortocircuito total automático de protección contra sobretensiones»)? Controla una tensión de bus particular y la desconecta si se hace superior a un nivel predeterminado. Los circuitos de OVP pueden diseñarse para dispararse a 1 mV por encima de nuestra banda de tolerancia del 5%. Dichos circuitos no sólo son complicados, sino que también pueden crear problemas adicionales por disparos accidentales. Los modos en que la avería es más probable que se produzca se refieren a un cortocircuito del regulador o unión accidental de dos buses, por ejemplo, los de +5 V y +12 V. En cualquier caso, el resultado es una elevación rápida de la tensión en las líneas de salida. Cuando la tensión se eleva por encima del valor de zener, la corriente circula por la puerta de SCR. En un cierto punto, que suele estar por debajo de donde se hubiera deteriorado cualquier componente, el SCR se excita y cortocircuita la línea de salida a masa. La corriente excesiva destruye el fusible, eliminando el regulador, o reguladores, que plantean problemas (ambos fusibles se destruirían si estuvieran conectadas las alimentaciones de +5 V y +12 V). Todo ello sucede muy rápido. El circuito de prueba de la figura 1.23 pone de manifiesto lo que sucede cuando el OVP de +5 V ha aplicado repentinamente +12 V. Los circuitos de

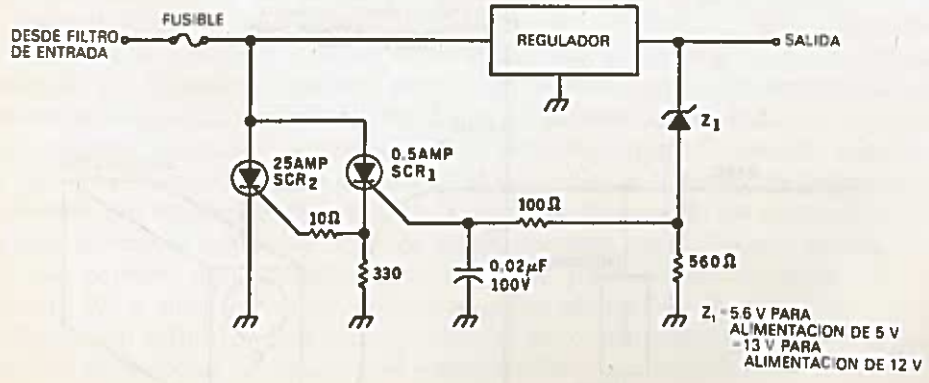


Figura 1.21 *Esquema de un circuito de protección contra sobretensiones más complejo. La sección de cortocircuito total del OVP puede estar situada cerca del fusible, mientras que el sensor de OVP Z_1 está ubicado en la salida del regulador. Esta es una colocación preferida de los componentes si el sensor y la fijación pueden separarse adecuadamente. El sensor de baja corriente Z_1 excita a SCR_1 en una condición de sobretensión. SCR_1, a su vez, excita a SCR_2 de alta corriente. La combinación de SCr_5 permite una libertad considerable en la elección de SCR_2, puesto que la cuestión de la corriente de puerta se hace menos importante.*

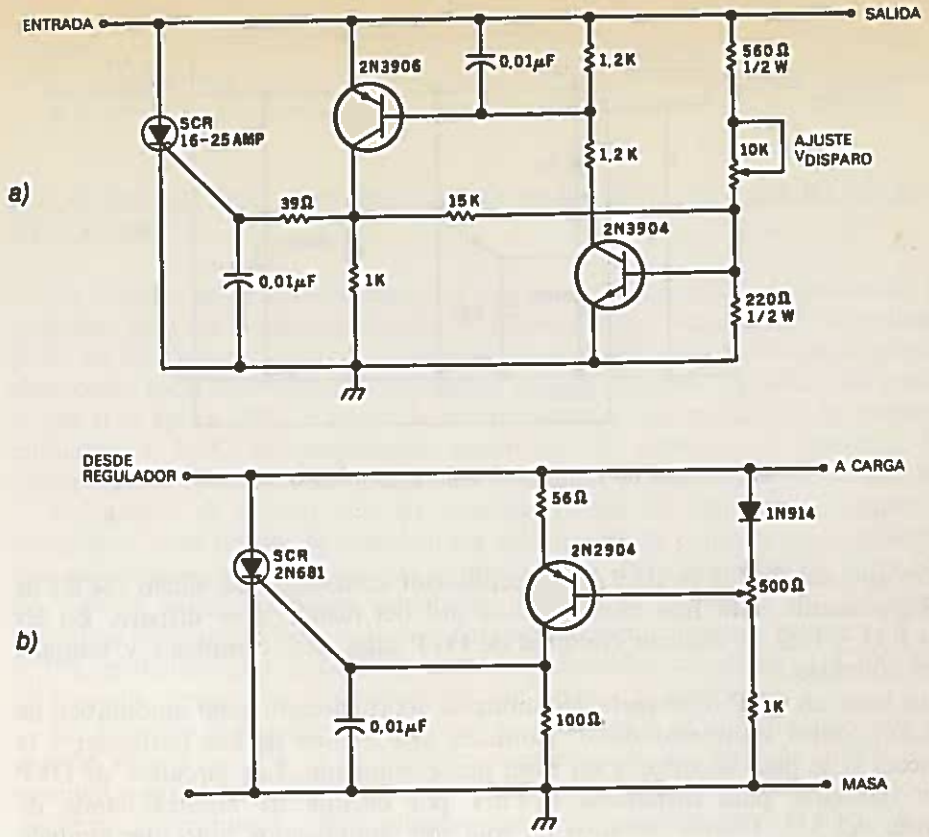


Figura 1.22 *Esquemas de circuitos de protección de sobretensiones con tensión ajustable.*
a) Un circuito OVP de tensión ajustable con un amplificador de corriente interna para excitar la puerta de SCR.
b) Un circuito alternativo para un circuito simple de OVP de tensión ajustable.

prueba son la única forma de poner de manifiesto la acción de un OVP. Si su fuente de alimentación funciona adecuadamente, nunca debe dispararse. El SRC nunca permite que la línea pase a 12 V antes de fijarle a masa. La sustitución del fusible por una resistencia de 22 ohmios permite aplicaciones múltiples del pulsador sin reemplazar fusibles.

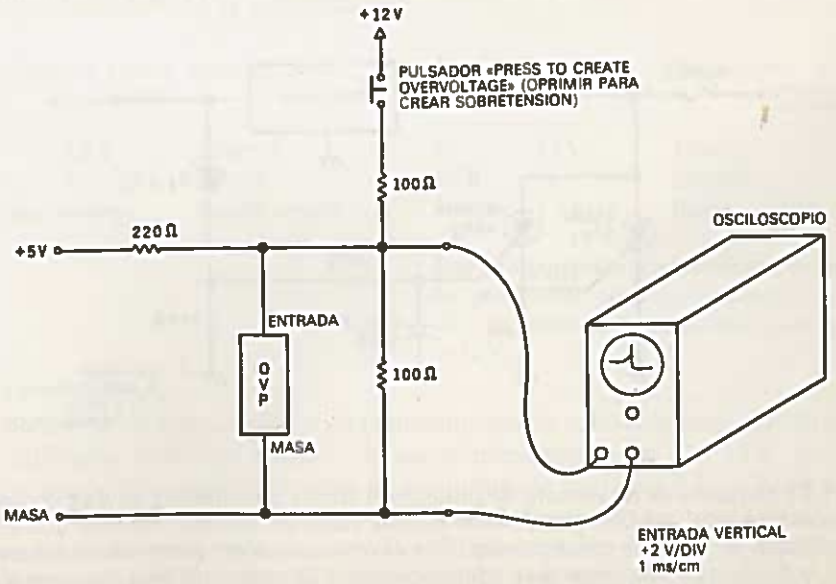


Figura 1.23 *Un circuito de prueba para poner de manifiesto la acción del protector contra sobretensiones.*

Capítulo 2

ELEMENTOS BASICOS DEL PROCESADOR CENTRAL

Hay muchos microprocesadores diferentes en el mercado y aunque la nomenclatura de las instrucciones es algo distinta para cada uno, los procesos básicos de cómputo lógico son semejantes en todos los dispositivos. La regla a recordar la siguiente vez que se discuta las posibilidades de dos computadoras es que «una computadora es una computadora». No podemos aseverar que son completamente idénticos estos dispositivos, pero son numerosas las similitudes y no nos gustaría perder tiempo en analizar los juegos de instrucciones y los detalles de interconexión antes de elegir uno.

En cierta ocasión, el autor tuvo la oportunidad de conversar con el diseñador de uno de los sistemas de computadoras personales de mayor venta en el mercado. Se habían vendido miles de computadoras, generando inmensos beneficios para el fabricante. Nuestra conversación se centró, a la larga, sobre la relación coste-prestaciones de su diseño. El autor tenía ideas fantásticas de un equipo de diseño, tardando meses en reducir el número de componentes y en analizar los juegos de instrucciones para determinar las necesidades mínimas de memoria. En ese preciso momento, se le había dado al diseñador un plazo de dos meses para conseguir un diseño con posibilidades de fabricación. Lo único que preocupaba a los inversores era el precio y la disponibilidad de los componentes particulares que había elegido el diseñador. Al ser un gran entusiasta de las computadoras personales, construyó simplemente una computadora sobre la base del microprocesador que ya poseía. La eventual publicidad de su sistema se basaba en la arquitectura avanzada que se materializaba en el procesador central, pero el usuario no tenía a su disposición medio alguno de programación en lenguaje-máquina. Tenía solamente un intérprete BASIC de lenguaje de alto nivel y, desde el punto de vista técnico, era simplemente una computadora de «caja negra». Podía haber utilizado cualquier otro microprocesador. Lo mismo que para un diseño de libro de texto.

Lamentablemente, el aficionado que está construyendo una microcomputadora «a partir de la nada», y que no desea hacer una «caja negra», ha de intentar conseguir un dispositivo que esté situado en la zona media del espectro de rendimientos y capacidades funcionales. La regla general de que todas las computadoras realizan funciones similares es cierta, pero una tarjeta de circuito impreso es un lujo. El aficionado que ha de hacer todo el cableado a mano, de seguro estará interesado por un diseño eficiente. En la realidad, algunos de los microprocesadores más esotéricos requieren circuitos periféricos muy caros. Incluso los dispositivos que parecen muy sencillos, con juegos de instrucciones limitados, pueden requerir 50, o más, circuitos integrados como elementos de interconexión. La configuración definitiva debe ser una solución de compromiso entre la complejidad circuital, la facilidad de prueba y el precio de los componentes.

ARQUITECTURA DEL MICROPROCESADOR

La arquitectura del microprocesador determina los dispositivos de soporte requeridos para hacer un sistema de microcomputadora. Quizá lo mejor sea comenzar por examinar brevemente las principales diferencias en la arquitectura.

Definición: Una microcomputadora es una máquina lógica que manipula números binarios (datos) y procesa esta información siguiendo una secuencia organizada de pasos de programa, a los que se les conoce como instrucciones.

Todas las microcomputadoras, como todas las computadoras, tienen las características siguientes:

1. **Entrada.**—Deben existir medios para permitir la introducción de datos o instrucciones.
2. **Memoria.**—La secuencia del programa debe almacenarse antes y después de la ejecución y debe disponerse de recursos para almacenar el resultado de cualquier cómputo.
3. **Unidad aritmética y lógica.**—Realiza operaciones aritméticas con los datos almacenados o de entrada.
4. **Sección de control.**—Toma decisiones con respecto al flujo del programa y control del proceso basándose en los estados internos de los resultados de cálculos aritméticos.
5. **Salida.**—Los resultados se entregan al usuario o se almacenan en un medio adecuado.

El microprocesador es el circuito integrado único sobre el que está construido una microcomputadora. *El microcomputador es un dispositivo; la microcomputadora es un sistema.* En su forma menos compleja, los microprocesadores sólo incluyen las funciones de tres y cuatro elementos de datos y deben contar con dispositivos externos unidos a los buses para realizar las demás tareas. En la figura 2.1 se ilustra el diagrama de bloques básico de una microcomputadora de 8 bits y muestra la interconexión de estos buses y elementos de soporte. La computadora de la figura 2.1 utiliza seis buses independientes: direcciones de memoria, datos de memoria (entrada y salida), direcciones de E/S y entrada y salida de datos. El microprocesador contiene un procesador central que está constituido por los circuitos requeridos para tener acceso a la memoria adecuada y las posiciones de E/S y para interpretar las instrucciones resultantes que también se ejecutan en esta unidad. El procesador central contiene también la ALU (unidad aritmética y lógica), que es una red combinacional que realiza operaciones aritméticas y lógicas con los datos. Además, el procesador central incluye una sección de control que rige el funcionamiento de la computadora y los diversos registros de datos utilizados para la manipulación y almacenamiento de datos e instrucciones.

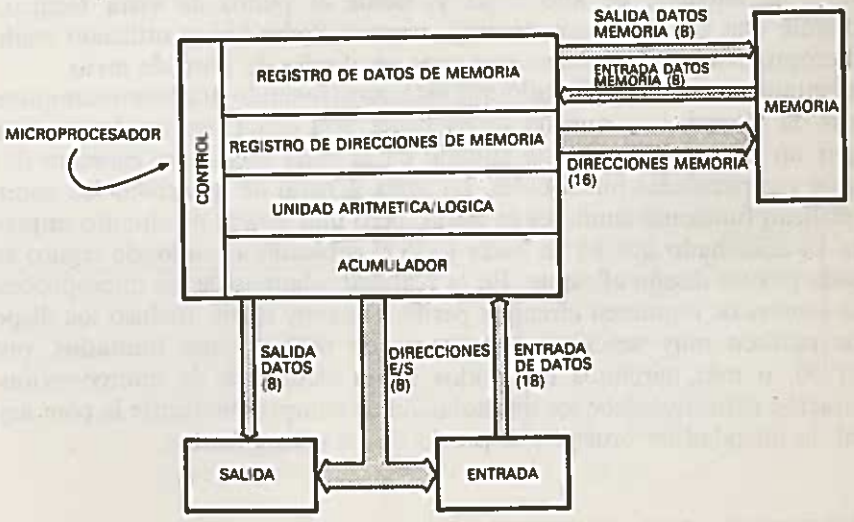


Figura 2.1 *Un diagrama de bloques básico de una microcomputadora que ilustra el concepto de la disposición de buses de datos. Los números entre paréntesis son la cantidad usual requerida de hilos físicos para realizar funciones de buses para un microprocesador de 8 bits.*

Realmente, pocos microprocesadores tienen seis buses independientes. El número de terminales que se requeriría en el circuito integrado es algo no trascendente. En cambio, para reducir las salidas de terminales, los fabricantes de componentes suelen combinar los buses de entrada y salida de datos y hacerlos «bidireccionales». Durante una instrucción de salida, los datos fluyen desde el microprocesador al dispositivo de salida, y viceversa, durante una instrucción de entrada. Para reducir todavía más el número de terminales requeridos en el procesador central, el bus de direcciones de memoria puede servir también como el bus de direcciones para dispositivos de entrada y salida. Durante las instrucciones de entrada/salida, la dirección presente en las líneas de dirección se refiere a un dispositivo de entrada/salida particular. En la figura 2.2 se muestra la configuración reducida resultante.

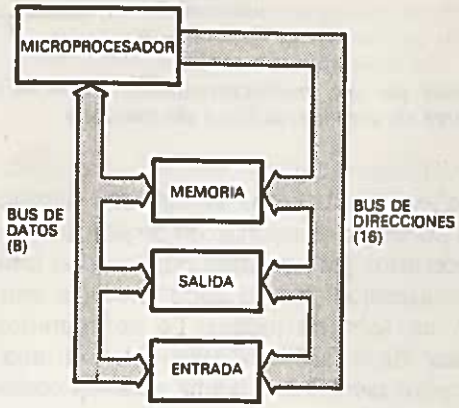


Figura 2.2 *Un diagrama de bloques de una microcomputadora que utiliza técnicas de configuración de buses bidireccionales para reducir las salidas de terminales.*

El concepto de dos buses es fácil de comprender y, desde un punto de vista del hardware, sencillo de utilizar. Los buses están multiplexados en tiempo y función. Es decir, durante las operaciones de memoria, los bits en el bus de direcciones se refieren a una posición de memoria y los datos en el bus de datos representan el contenido de memoria. El sentido del flujo de datos (a o desde el procesador central) está controlado dentro del microprocesador. Las actividades con dispositivos de entrada/salida se realizan de una forma similar. Durante esas instrucciones, los datos de entrada o de salida y las direcciones de dispositivos ocupan los buses.

El número de hilos de conexión de buses puede reducirse más combinando los datos y las direcciones en las mismas líneas y multiplexando en tiempo la transferencia de datos a lo largo de ellas. En la figura 2.3 se ilustra esta configuración final. Este método requiere elementos circuitales adicionales para desmultiplexar y almacenar los datos pertinentes. Los componentes externos adicionales necesarios para utilizar esta característica de la arquitectura hacen que su uso no sea recomendable para el aficionado. Hay otros microprocesadores que son más sencillos de utilizar.

Cuando se construye, más bien que adquirir, una computadora personal, deben considerarse detenidamente los criterios siguientes:

1. **Complejidad del circuito.**—Mantener los componentes en un mínimo razonable. Cuantos más componentes hay en un diseño, tanta más probabilidad existe de errores de cableado y de dispositivos defectuosos.
2. **Coste.**—Aunque el coste sea importante, no debe ser la consideración primaria. Cualquier función de microprocesador podría simularse con el empleo de lógica de integración a pequeña escala; sin embargo, los costes indirectos que resultan de utilizar 200 pastillas integradas para sustituir 3 ó 4 dispositivos LSI (integración a gran escala) descalificarían el valor de emplear inicialmente componentes más baratos. Por el contrario, en la

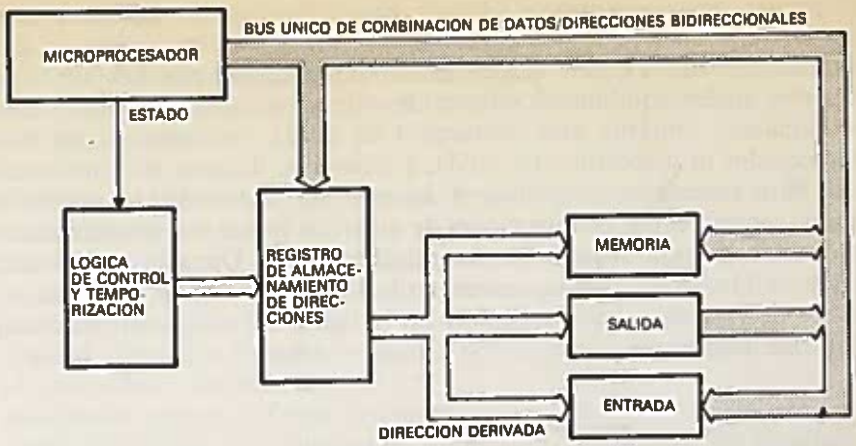


Figura 2.3 *Diagrama de bloques de una microcomputadora que utiliza un bus bidireccional multiplexado único para funciones de entrada/salida y de memoria.*

industria de los semiconductores, la densidad significa mayor coste. Cuantas más funciones puede proporcionar un dispositivo, y cuantos menos componentes sean necesarios para realizar estas tareas, tanto mayor será el precio. El nivel de integración incorporado en una computadora casera debe mantenerse en un término medio. La computadora ZAP es un ejemplo notorio de esta filosofía de diseño. Utiliza una combinación de LSI (integración a gran escala) con buena relación coste-eficacia y SSI (integración a pequeña escala) no cara para obtener una computadora que el aficionado pueda construir, probar y utilizar de forma efectiva.

3. **Compatibilidad y disponibilidad de software.**—La construcción del hardware de una microcomputadora es sólo la mitad del trabajo. Debe estar programada para realizar un trabajo útil. Inicialmente, el constructor, por necesidad, manejará códigos y ensamblará sus propios programas. Sin embargo, a la larga, puede surgir la necesidad de que la computadora realice una tarea que requiera un programa muy grande que no pueda ensamblarse fácilmente. El usuario debe contar con un programa ensamblador en una máquina más grande. Por supuesto, el programa ensamblador tendría que ser compatible con el juego de instrucciones de la microcomputadora.

   Una consideración adicional es que los entusiastas de las computadoras personales están siempre intercambiando software. Es posible convertir programas para que se puedan ejecutar en cualquier procesador central, pero el esfuerzo sería el mismo que escribir el programa completo a partir de la nada. Ello niega la necesidad de intercambiar software. El propietario de la computadora debe elegir un microprocesador que sea algo compatible con las computadoras existentes en el mercado. La aseveración que hicimos de que todas las computadoras son semejantes es teóricamente cierta, pero un libro sobre cómo construir una computadora esotérica, única en su género, es de poco valor práctico.

Cada criterio podría analizarse y replicarse individualmente, pero hemos de dar algún crédito a los fabricantes de computadoras personales que ya hicieron lo mismo por nosotros. El hecho de que tantas computadoras personales estén en uso ha establecido «de facto» la normalización de la elección de procesador central. Para ser compatible con el software existente y tener suficiente documentación disponible, el constructor debe considerar la elección entre los procesadores centrales en uso en el comercio. Los cuatro microprocesadores más utilizados son:

1. Intel 8080A
2. Motorola 6800
3. MOS Technology 6502
4. Zilog Z80

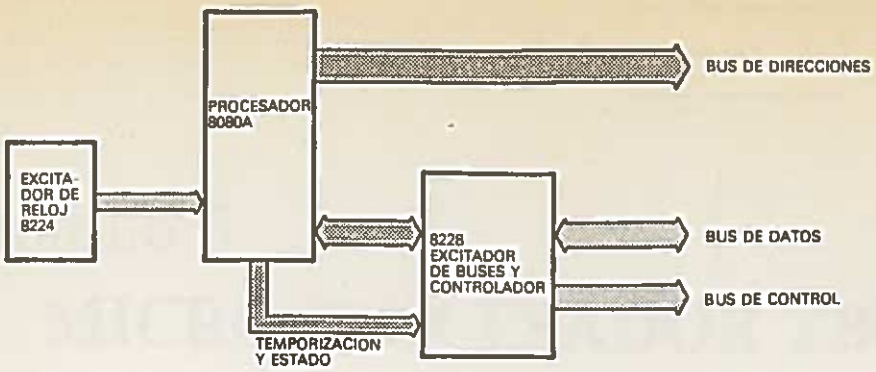


Figura 2.4 *Configuración mínima de tres pastillas integradas del 8080A que ilustra los dispositivos de apoyo necesarios. El bus de control contiene las funciones de temporización necesarias para decodificar el contenido de los buses de datos y de direcciones.*

Como consecuencia de la gran difusión de cada dispositivo, la documentación y el software pueden obtenerse fácilmente. La disponibilidad de software compatible con el 8080A es más grande y el coste es bajo, pero su complejidad circuital es también la mayor de todos los microprocesadores antes citados. El 8080A, aunque se describe como «una computadora en una sola pastilla», cuenta con varios excitadores-controladores externos y dispositivos de apoyo. Su configuración funcional mínima está constituida por tres pastillas integradas, como se muestra en la figura 2.4. Su estructura de buses del procesador central es similar a la que se ilustra en la figura 2.3, pero cuando se combina con las pastillas soporte 8224 y 8228, emula la arquitectura de buses más deseable indicada en la figura 2.2.

Lo mejor de ambos mundos está incorporado dentro del Z80. No solamente ejecuta el juego completo de instrucciones del 8080A, sino que también tiene instrucciones adicionales que sirven para hacerle un procesador muy poderoso. En la figura 2.5 se ilustra la estructura de buses del Z80. El Z80 es algo más caro que los otros procesadores citados. Sin embargo, su circuitería externa reducida da lugar a una comparación efectiva de los costes. Asimismo, la facilidad de interconexión del Z80 le hace ser la elección natural cuando se construye una microcomputadora a partir de nada ya existente.

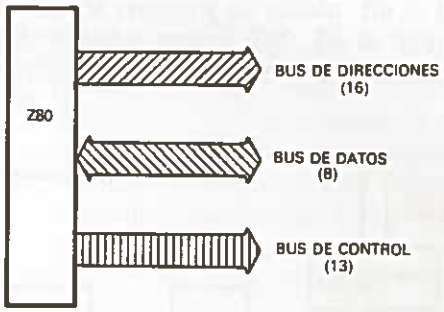


Figura 2.5 *Diagrama de bloques de la estructura de buses del Zilog Z80.*

Capítulo 3
EL MICROPROCESADOR Z80

Muchos libros se han escrito sobre las propiedades de software y de hardware del Z80. Aunque el autor no pretende duplicar los esfuerzos de otros, cualquier libro destinado a la construcción de una microcomputadora sería incompleto sin una sección que describa el procesador con algún detalle. Comprendiendo a fondo la lógica interna y las funciones de control externas del procesador central, será capaz de entender mejor la forma en que hemos desarrollado el resto del hardware del sistema. Tendrá muchas opciones cuando construya una computadora no partiendo de nada existente. Cuanto más profundo sea su conocimiento del microprocesador, tanto mayor será su confianza en el resultado a conseguir y tanto más probable será que haga mejoras de su propio diseño original.

La computadora ZAP permite que se le conecten una amplitud considerable de periféricos. La elección depende principalmente de la filosofía de diseño del sistema, que comienza con el procesador central.

ARQUITECTURA DEL PROCESADOR CENTRAL

El Z80 es un microprocesador orientado hacia los registros. Dieciocho registros de 8 bits y cuatro registros de 16 bits dentro del procesador central son accesibles al programador y funcionan como memoria programable estática. Estos registros se dividen en dos conjuntos, principal y alternativo, cada uno de los cuales contiene seis registros de 8 bits de uso general, que pueden utilizarse bien sea individualmente, bien sea como tres pares de registros de 16 bits. También incluyen dos conjuntos de acumuladores y de registros de estado. En la figura 3.1 se ilustra la arquitectura interna del procesador central Z80. En la figura 3.2 se muestra que dentro del Z80 hay acumuladores y registros de estado, junto con registros generales y de uso especial.

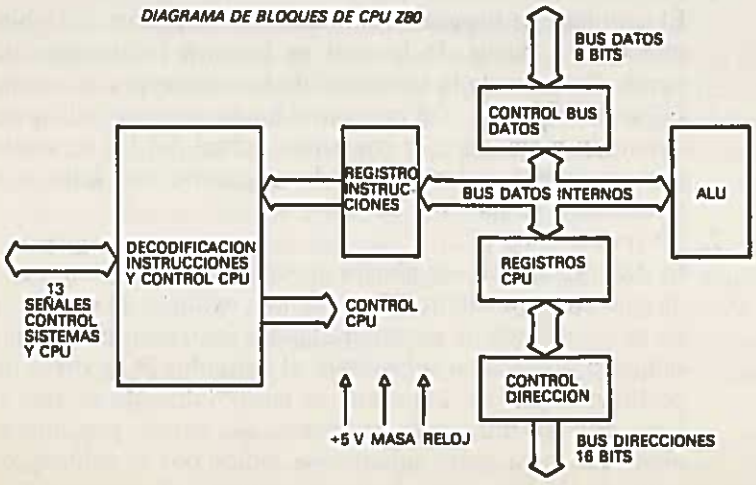


Figura 3.1 *Diagrama de bloques de la arquitectura interna del procesador central Z80.*

Figura 3.2 *Configuración de registros de procesador central Z80.*

A continuación se da una descripción de la función y de la estructura de los componentes principales del procesador central.

I. Registros
   A. Acumuladores y registros de estado
      El procesador central contiene dos pares independientes de acumulador y registro de estado, uno en el conjunto de registros principales y el otro en el conjunto de registros alternativos. El acumulador recibe los resultados de todas las operaciones lógicas y aritméticas de 8 bits, mientras que el registro de estado indica la ocurrencia de condiciones específicas lógicas o aritméticas en el procesador tales como paridad, cero, signo, acarreo y desbordamiento. Una instrucción de intercambio simple permite al programador seleccionar uno u otro par de acumuladores o registros de estado.
   B. Registros de uso general
      Hay dos conjuntos similares de registros de uso general. El conjunto de registros principales contiene seis registros de 8 bits llamados B, C, D, E, H y L; el conjunto de registros alternativos contiene también seis registros de 8 bits que se denominan B', C', D', E', H' y L'. Para operaciones de 16 bits, estos registros pueden agruparse en pares de 16 bits (BC, DE, HL o BC', DE', HL'). Una instrucción de intercambio único permite al programador elegir alternativamente entre los conjuntos de pares de registros.
   C. Registros de uso especial
      1. PC (contador de programa)
         El contador de programa contiene una dirección de 16 bits en memoria a partir de la cual se buscará la instrucción en curso. Después de la ejecución de la instrucción, el contador PC se incrementa, si el programa ha de proseguir al siguiente octeto en memoria o el contenido actual del PC se sustituye por un nuevo valor, si ha de ejecutarse un salto o una instrucción de llamada.
      2. SP (puntero de pila)
         El Z80 permite varios niveles de «anidación» (jerarquización) de subrutinas mediante el uso de una «pila» y de un «puntero de pila»: cuando se ejecutan algunas instrucciones o cuando se hacen llamadas a subrutinas, el contador PC y otros datos pertinentes pueden almacenarse temporalmente en una pila. Una pila es una zona reservada de varias posiciones de memoria, cuya parte superior se indica por el contenido del puntero de pila. Es decir, el puntero de pila indica la dirección de la entrada hecha más recientemente, porque las

posiciones de memoria están organizadas como un fichero de «último en entrar, primero en salir». Al examinar las entradas particulares en la pila, el procesador central retorna a un programa principal haciendo caso omiso de la profundidad de las subrutinas «anidadas» (jerarquizadas) teóricamente, la pila podría tener una longitud de 64 koctetos; sin embargo, el espacio de programa no debe recubrirse en escritura mediante una pila en expansión.

D. Registros índice IX y IY
Estos registros facilitan la manipulación de datos de tablas. Son dos registros independientes de 16 bits, que retienen las direcciones de base utilizadas en modos de direccionamiento indexado y apuntan a posiciones en memoria, cuando han de almacenarse o recuperarse datos pertinentes. Incorporado en las instrucciones indexadas hay un entero con signo en complemento a 2, que especifica el desplazamiento desde esta dirección de base.

E. Registro de direcciones de página de interrupción (I)
Este es un registro de 8 bits que puede cargarse con una dirección de página de una rutina de servicio de interrupción. Durante un programa de interrupción modo 2, el control vectorizará esta dirección de página.

F. Registro de regeneración («refresco») de memoria (R)
Para validar memorias dinámicas para el Z80, un registro de regeneración de memoria de 7 bits se incrementa automáticamente después de cada búsqueda de instrucción.

II. Unidad lógica y aritmética
Las manipulaciones aritméticas y las operaciones lógicas se tratan como 8 bits simultáneos en la ALU (unidad lógica y aritmética) del Z80. La ALU se comunica internamente con los registros del procesador central y no es directamente accesible por el programador. La ALU realiza las operaciones siguientes:

DESPLAZAMIENTO A IZQUIERDA o A DERECHA
INCREMENTO
DECREMENTO
SUMA
RESTA
AND
OR
OR EXCLUSIVA
COMPARACION
PUESTA A UNO DE BIT
PUESTA A CERO DE BIT
PRUEBA DE BITS

III. Registro de instrucciones y control de procesador central
El registro de instrucciones retiene el contenido de la posición de memoria direccionada por el PC (contador de programa) y se carga durante el ciclo de búsqueda de cada instrucción. La unidad de control del procesador central ejecuta las funciones definidas por la instrucción en el registro de instrucciones y genera todas las señales de control necesarias para transmitir los resultados a los registros adecuados.

IV. Hardware del procesador central

A. En la figura 3.3 se detalla la disposición de las salidas (diagrama de conexiones) de los terminales del Z80. Se suministra en un encapsulado normalizado DIL de 40 terminales. A continuación se da una lista y la explicación correspondiente de las funciones de los terminales.

A_0-A_{15} (bus de direcciones)	Salida triestado, activa a nivel alto. A_0-A_{15} constituyen un bus de direcciones de 16 bits. Estas señales proporcionan la dirección correspondiente a intercambios de datos de memoria (hasta 64 koc-

tetos) y a los intercambios de datos de dispositivos de E/S. En el direccionamiento de E/S se utilizan los ocho bits de dirección inferiores para permitir al usuario seleccionar directamente hasta 256 ports de entrada o 256 ports de salida. A_0 es el bit de dirección menos significativo. Durante el tiempo de regeneración, los siete bits inferiores contienen una dirección de refrescación válida.

D_0-D_7 (Bus datos) — Entrada/salida triestado, activa a nivel alto. D_0-D_7 constituyen un bus de datos bidireccional de 8 bits, que se utiliza para intercambios de datos con memoria y dispositivos de E/S.

$\overline{M1}$ (ciclo de máquina 1) — Salida, activa a nivel bajo. $\overline{M1}$ indica que el ciclo de máquina en curso es el ciclo de búsqueda del código de operación de una ejecución de instrucción. Ha de tenerse presente que durante la ejecución de códigos de operación de 2 octetos, $\overline{M1}$ es generada cuando se busca cada octeto de código de operación. Estos códigos de operación de 2 octetos comienzan siempre con CBH, DDH, EDH o FDH. $\overline{M1}$ se presenta también con $\overline{IORQ}$ para indicar un ciclo de acuse de recibo de interrupción.

$\overline{MREQ}$ (petición de memoria) — Salida triestado, activa a nivel bajo. La señal de petición de memoria indica que el bus de direcciones retiene una dirección válida para una operación de lectura de memoria o de escritura de memoria.

$\overline{IORQ}$ (petición de entrada/salida) — Salida triestado, activa a nivel bajo. Esta señal indica que la mitad inferior del bus de direcciones retiene una dirección de E/S válida para una operación de lectura o de escritura de E/S. Una señal $\overline{IORQ}$ se genera también con una señal $\overline{M1}$, cuando se acusa recibo de una interrupción, para indicar que un vector de respuesta de interrupción puede colocarse en el bus de datos. Operaciones de acuse de recibo de interrupción pueden producirse durante el tiempo de $\overline{M1}$, aunque están prohibidas las operaciones de E/S.

$\overline{RD}$ (lectura de memoria) — Salida triestado, activa a nivel bajo. $\overline{RD}$ indica que el procesador central desea la lectura de memoria o de un dispositivo de E/S. La memoria, o dispositivo de E/S direccionado, debe utilizar esta señal para controlar los datos en el bus de datos del procesador central.

$\overline{WR}$ (escritura de memoria) — Salida triestado, activa a nivel bajo. $\overline{WR}$ indica que el bus de datos del procesador central retiene datos válidos a almacenarse en la memoria o dispositivo de E/S objeto de direccionamiento.

$\overline{RFSH}$ (regeneración) — Salida, activa a nivel bajo. $\overline{RFSH}$ indica que los siete bits inferiores del bus de direcciones contienen una dirección de regeneración para memorias dinámicas y la señal $\overline{MREQ}$ en curso debe emplearse para realizar una lectura de regeneración para todas las memorias dinámicas.

$\overline{HALT}$ (estado de parada) — Salida, activa a nivel bajo. $\overline{HALT}$ indica que el procesador central ha ejecutado una instrucción HALT y está esperando una interrupción no enmascarable o enmascarable (con la máscara validada) antes de que pueda reanudarse la operación. Mientras se esta en esta condición, el procesador

central ejecuta NOPs (sin operación) para mantener la actividad de regeneración de memoria.

$\overline{\text{WAIT}}$ (espera) — Entrada, activa a nivel bajo. $\overline{\text{WAIT}}$ indica al procesador central Z80 que los dispositivos de E/S o memoria, objeto de direccionamiento, no están preparados para una transferencia de datos. El procesador central continúa introduciendo estados de espera mientras que $\overline{\text{WAIT}}$ es activa; esta señal permite que la memoria de dispositivos de E/S se sincronice con el procesador central.

$\overline{\text{INT}}$ (interrupción) — Entrada, activa a nivel bajo. La señal de petición de interrupción se genera por dispositivos de E/S. Una petición se aceptará al final de la instrucción en curso si está activado el flip-flop de validación de interrupción bajo control de software interno y si no es activa la señal $\overline{\text{BUSRQ}}$. Cuando el procesador central acepta la interrupción, una señal de acuse de recibo ($\overline{\text{IORQ}}$ durante el tiempo $\overline{\text{M1}}$) se envía al principio del siguiente ciclo de interrupción. El procesador central puede responder a una interrupción en los tres modos diferentes.

$\overline{\text{NMI}}$ (interrupción no enmascarable) — Entrada, disparada por flanco negativo. La línea de petición de interrupción no enmascarable tiene una prioridad más alta que $\overline{\text{INT}}$ y siempre se identifica al final de la instrucción en curso, prescindiendo del estado del flip-flop de validación de interrupción. $\overline{\text{NMI}}$ obliga al procesador central Z80 a reiniciarse en la posición 0066_{16}. El contador de programa se conserva automáticamente en la pila externa, de modo que el usuario pueda volver al programa que fue interrumpido. Ha de tenerse en cuenta que los ciclos de espera (WAIT) continuos pueden impedir que finalice la instrucción en curso y que una señal $\overline{\text{BUSRQ}}$ anulará una señal $\overline{\text{NMI}}$.

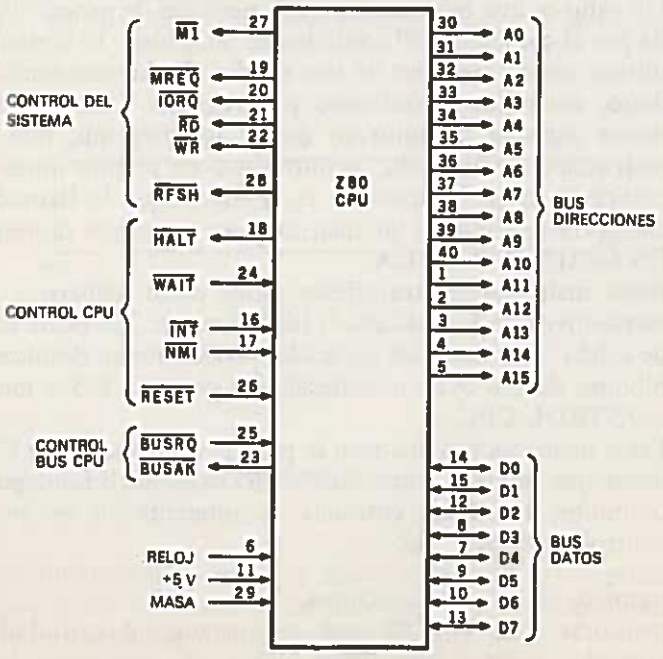


Figura 3.3 *Configuración de los terminales (pines) para el microprocesador Z80.*

La temporalización real de estas señales, se examinará en las secciones del hardware.

V. Tipos de instrucciones del Z80
El Z80 puede ejecutar 158 instrucciones independientes incluyendo las 78 instrucciones del 8080A.
Puede agruparse como sigue:

A. CARGA E INTERCAMBIO
Las instrucciones de carga desplazan datos entre registros o entre registros y memoria. La fuente y el destino de estos datos se especifican dentro de la instrucción. Las instrucciones de intercambio permiten el contenido de dos registros.

B. ARITMETICAS Y LOGICAS
Estas instrucciones actúan sobre los datos en el acumulador, un registro o una posición de memoria designada. Los resultados se colocan en el acumulador y los indicadores de estado se establecen consecuentemente. Las operaciones aritméticas incluyen la adición y resta de 16 bits entre pares de registros.

C. BUSQUEDA Y TRANSFERENCIA DE BLOQUES
El Z80 utiliza una sola instrucción para transferir cualquier bloque de memoria a cualquier otro grupo de posiciones de memoria contiguas. La búsqueda de bloques utiliza una sola orden para examinar un bloque de memoria para un carácter de 8 bits particular.

D. ROTACION Y DESPLAZAMIENTO
Los datos pueden ser objeto de rotación y desplazamiento en el acumulador, un registro del procesador central o memoria. Estas instrucciones tienen también medios de tratamiento del código decimal codificado en binario (BCD).

E. MANIPULACION DE BITS
La manipulación de bits incluye las funciones de posicionamiento, de inicialización y de prueba. Los bits individuales pueden modificarse o probarse en el acumulador, un procesador central o memoria. Los resultados de las operaciones de prueba se indican en el registro de estado.

F. SALTO, LLAMADA Y RETORNO
Un salto es una bifurcación a una posición de programa especificada por el contenido del contador de programa. El contenido de este último puede proceder de tres modos de direccionamiento: inmediato, extendido o indirecto por registro. Una llamada es una forma especial de salto en donde la dirección, que sigue a la instrucción de llamada, se introduce en la pila antes de que se realice el salto. Un retorno es la inversa de la llamada. En esta categoría se incluyen las instrucciones especiales de reinicio.

G. ENTRADA Y SALIDA
Estas instrucciones transfieren datos entre registro y memoria a dispositivos de E/S externos. Se dispone de 256 ports de entrada y de salida. Instrucciones especiales proporcionan desplazamiento de bloques de 256 bytes a, o desde, los ports de E/S y memoria.

H. CONTROL CPU
Estas instrucciones incluyen la parada funcional de la CPU o la de hacer que se ejecute una NOP (sin operación). La capacidad para desinhibir o inhibir entradas de interrupción es un medio de control suplementario.

VI. Formatos de datos e instrucciones
La memoria para el Z80 está organizada en cantidades de 8 bits denominados octetos (ver figura 3.4). Cada octeto de programa está almacenado en una posición de memoria singular y es objeto de referencia por medio de una dirección binaria de 16 bits.

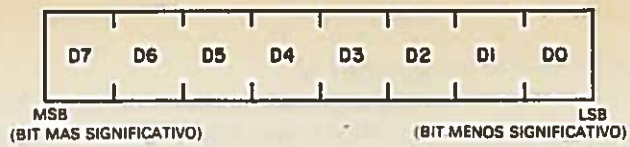


Figura 3.4 *Organización de un octeto de datos en el Z80.*

La capacidad de direccionamiento directo total es de 65 536 octetos (64 K) de memoria, que puede ser cualquier combinación de ROM (memoria de sólo lectura); EPROM (memoria de sólo lectura borrable-programable) o memoria programable. Los datos se almacenan en los formatos de la figura 3.5.

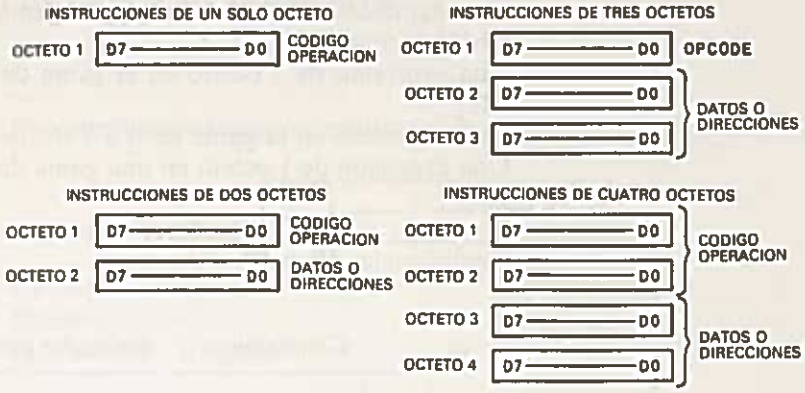


Figura 3.5 *Formatos de instrucciones en lenguaje máquina para el Z80.*

VII. Indicadores de estado del Z80
El registro de estado (F y F') suministra información al usuario con respecto al estado del procesador central en cualquier momento dado. Hay cuatro bits de estado comprobables y dos no comprobables, en cada registro. En la figura 3.6 se muestra la posición y la identidad de estos bits de estado.

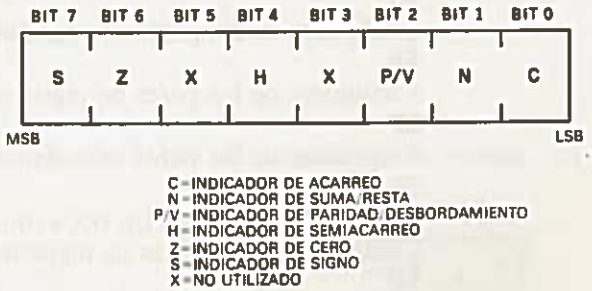


Figura 3.6 *Posiciones e identidad de los bits de indicación de estado en el registro de estado.*

Las instrucciones ponen a uno (bit estado=1) o ponen a cero (bit estado=0) los indicadores de una manera en correspondencia con la operación particular que se está ejecutando.

VIII. El juego de instrucciones del Z80
Los siguientes símbolos y abreviaturas se utilizan en la subsiguiente descripción de las instrucciones del Z80:

Símbolo	Significado
acumulador	Registro A
dirección	Una cantidad de dirección de 16 bits
dirección de orden alto	Los 8 bits más significativos de la dirección de 16 bits
dirección de orden bajo	Los 8 bits menos significativos de la dirección de 16 bits
datos	Una cantidad de 8 o de 16 bits
datos de orden alto	Los 8 bits más significativos de los datos de 16 bits
datos de orden bajo	Los 8 bits menos significativos de los datos de 16 bits
port	Una dirección de 8 bits de un dispositivo de E/S
r, r'	Uno de los registros A, B, C, D, E, H o L
n	Una expresión de 1 octeto en la gama de 0 a 255 inclusive
nn	Una expresión de 2 octetos en la gama de 0 a 65 535 inclusive
d	Una expresión de 1 octeto en la gama de −128 a 127
b	Una expresión en la gama de 0 a 7 inclusive
e	Una expresión de 1 octeto en una gama de −126 a 129
cc	El estado de los indicadores para instrucciones condicionales JR y JP:

cc	*Condición*	*Indicador pertinente*
000	NZ no cero	Z
001	Z cero	Z
010	NC no acarreo	C
011	C acarreo	C
100	PO paridad impar	P/V
101	PE paridad par	P/V
110	P signo positivo	S
111	M signo negativo	S

XXH	Indica valor de dirección hexadecimal
qq	Cualquiera de los pares de registros BC, DE, HL o AF
ss	Cualquiera de los pares de registros BC, DE, HL o SP
pp	Cualquiera de los pares de registros BC, DE, IX o SP
rr	Cualquiera de los pares de registros BC, DE, IY o SP
s	Cualquiera de r, n (HL), (IX+d) o (IY+d)
dd	Cualquiera de los pares de registros BC, DE, HL o SP
m	Cualquiera de r, (HL), (IX+d) o (IY+d)
(HL)	Especifica el contenido de memoria en la posición direccionada por el contenido del par de registros HL
(nn)	Especifica el contenido de memoria en la posición direccionada por la expresión de 2 octetos en nn
PC	Contador de programa
SP	Puntero de pila
t	Una expresión en la gama de 0 a 7 inclusive
C, N, P/V, H, Z, S	Indicadores de condición:

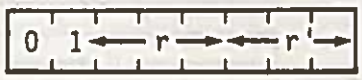
	C	Acarreo
	N	Suma/resta
	P/V	Paridad/desbordamiento
	H	Semiacarreo
	Z	Cero
	S	Signo
←		«se transfiere a»
∧		AND lógica
⊕		OR exclusiva
∨		OR inclusiva
+		Suma
−		Resta
↔		«se intercambia con»

GRUPO DE CARGA DE OCHO BITS

LD r, r'

r←r'

El contenido de cualquier registro r' se carga en cualquier otro registro r.



Ciclos: 1
Estados: 4
Indicadores: ninguno

LD r, n

r←n

El entero de 8 bits n se carga en cualquier registro r.

Ciclos: 2
Estados: 7
Indicadores: ninguno

LD r, (HL)

r←(HL)

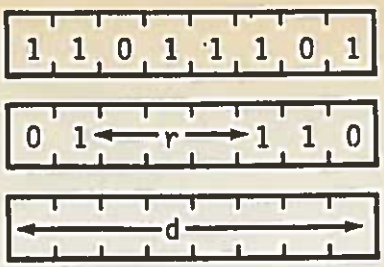
El contenido de 8 bits de posición de memoria (HL) se carga en el registro r.

Ciclos: 2
Estados: 7
Indicadores: ninguno

LD r, (IX+d)

r←(IX+d)

El operando (IX+d) (el contenido del registro índice IX sumado con un entero de desplazamiento d) se carga en el registro r.

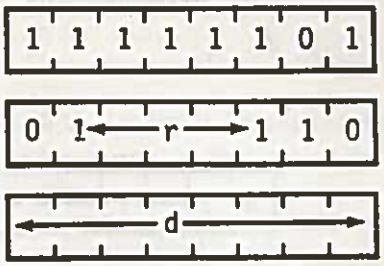


Ciclos: 5
Estados: 19
Indicadores: ninguno

LD r, (IY+d)

r←(IY+d)

El operando (IY+d) (el contenido del registro índice IY sumado con un entero de desplazamiento d) se carga en el registro r.



Ciclos: 5
Estados: 19
Indicadores: ninguno

LD (HL), r

(HL)←r

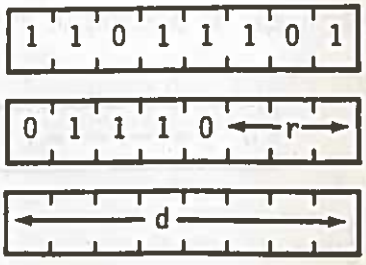
El contenido del registro r se carga en la posición de memoria especificada por el contenido del par de registros HL.

Ciclos: 2
Estados: 7
Indicadores: ninguno

LD (IX+d), r

(IX+d)←r

El contenido del registro r se carga a la dirección de memoria especificada por el contenido del registro IX sumado con d, que es un entero de desplazamiento de complemento a 2.



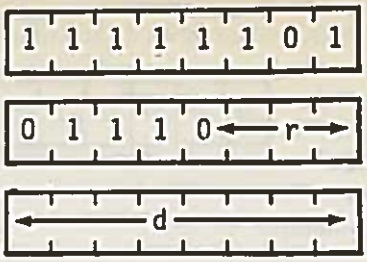
Ciclos: 5
Estados: 19
Indicadores: ninguno

LD (IY+d), r

(IY+d)←r

El contenido del registro r se carga en la dirección de memoria especifica-

da por la suma del contenido del registro índice IY y d, un entero de desplazamiento de complemento a 2.

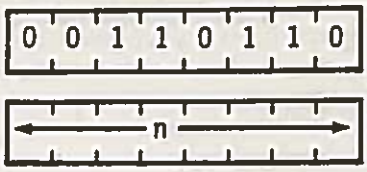


Ciclos: 5
Estados: 19
Indicadores: ninguno

LD (HL), n

(HL)←n

El entero n se carga en la dirección de memoria especificada por el contenido del par de registros HL.

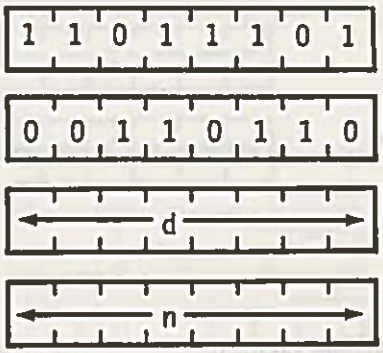


Ciclos: 3
Estados: 10
Indicadores: ninguno

LD (IX+d), n

(IX+d)←n

El operando n se carga en la dirección de memoria especificada por la suma del contenido del registro índice IX y el operando de desplazamiento, de complemento a 2, d.

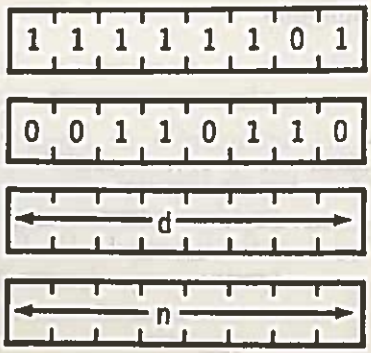


Ciclos: 5
Estados: 19
Indicadores: ninguno

LD (IY+d), n

(IY+d)←n

El entero n se carga en la posición de memoria especificada por el contenido del registro índice IY sumado con un entero de desplazamiento d.



Ciclos: 5
Estados: 19
Indicadores: ninguno

LD A, (BC)

A←(BC)

El contenido de la posición de memoria especificada por el contenido del par de registros BC se carga en el acumulador.

Ciclos: 2
Estados: 7
Indicadores: ninguno

LD A, (DE)

A←(DE)

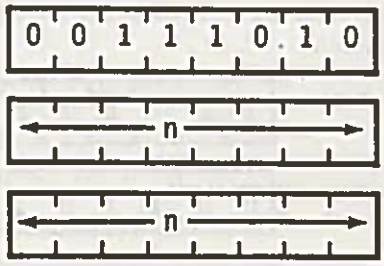
El contenido de la posición de memoria especificada por el par de registros DE se carga en el acumulador.

Ciclos: 2
Estados: 7
Indicadores: ninguno

LD A, (nn)

A←(nn)

El contenido de la posición de memoria especificada por los operandos nn se carga en el acumulador. El primer operando n es el octeto de orden bajo de una dirección de memoria de 2 octetos.



Ciclos: 4
Estados: 13
Indicadores: ninguno

LD (BC), A

(BC)←A

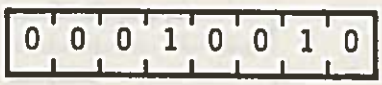
El contenido del acumulador se carga en la posición de memoria especificada por el contenido del par de registros BC.

Ciclos: 2
Estados: 7
Indicadores: ninguno

LD (DE), A

(DE)←A

El contenido del acumulador se carga en la posición de memoria especificada por el par de registros DE.

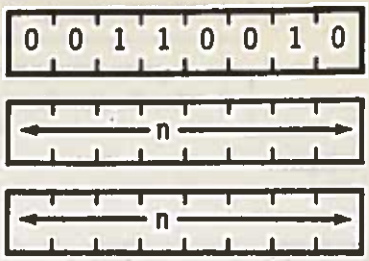


Ciclos: 2
Estados: 7
Indicadores: ninguno

LD (nn), A

(nn)←A

El contenido del acumulador se carga en la posición de memoria especificada por los operandos nn. El primer operando n es el octeto de orden bajo del operando nn.

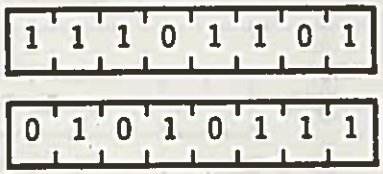


Ciclos: 4
Estados: 13
Indicadores: ninguno

LD A, I

A←I

El contenido del registro del vector de interrupción I se carga en el acumulador.



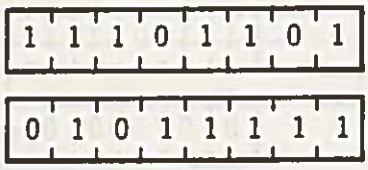
Ciclos: 2
Estados: 9
Indicadores: S, Z, H, N, P/V

S: poner a «1» si I<0; poner a «0» en cualquier otro caso
Z: poner a «1» si I=0; poner a «0» en cualquier otro caso
H, N: poner a «0»
P/V: retiene el contenido de IFF2

LD A, R

A←R

El contenido del registro de regeneración de memoria R se carga en el acumulador.



Ciclos: 2
Estados: 9
Indicadores: S, Z, H, N, P/V

S: poner a «1» si R<0; poner a «0» en cualquier otro caso
Z: poner a «1» si R=0; poner a «0» en cualquier otro caso
H, N: poner a «0»
P/V: incluye el contenido de IFF2

LD I, A

I←A

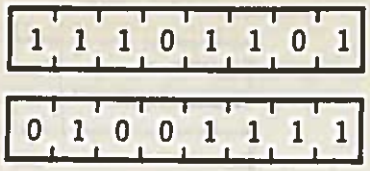
El contenido del acumulador se carga en el registro de vector de control de interrupción I.

Ciclos: 2
Estados: 9
Indicadores: ninguno

LD R, A

R←A

El contenido del acumulador se carga en el registro de regeneración de memoria R.



Ciclos: 2
Estados: 9
Indicadores: ninguno

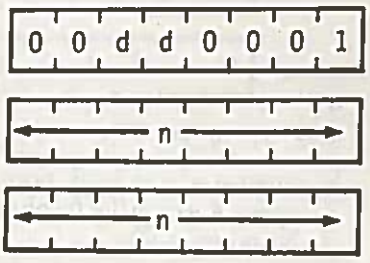
INSTRUCCIONES DE CARGA DE 16 BITS

LD dd, nn

dd←nn

El entero de 2 octetos nn se carga en el par de registros dd, en donde dd define los pares de registros BC, DE, HL o SP, ensamblados como se indica a continuación en el código objeto:

Par	*dd*
BC	00
DE	01
HL	10
SP	11

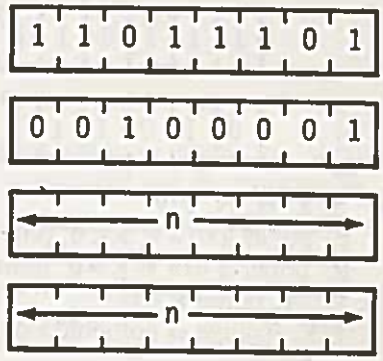


Ciclos: 3
Estados: 10
Indicadores: ninguno

LD IX, nn

IX←nn

El entero nn se carga en el registro índice IX.

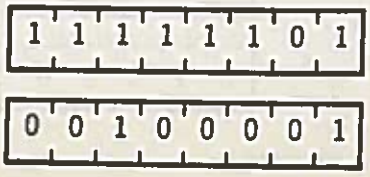


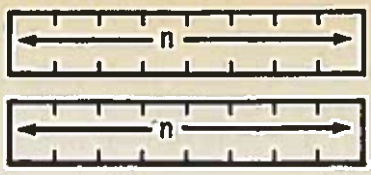
Ciclos: 4
Estados: 14
Indicadores: ninguno

LD IY, nn

IY←nn

El entero nn se carga en el registro índice IY.



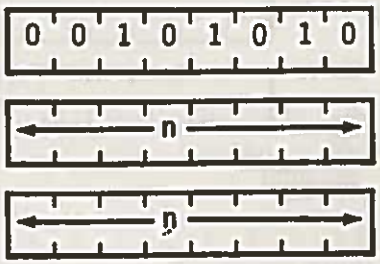


Ciclos: 4
Estados: 14
Indicadores: ninguno

LD HL, (nn)

H←(nn+1), L←(nn)

El contenido de la posición de memoria nn se carga en el registro L y el contenido de la siguiente posición de memoria más alta (nn+1) se carga en el registro H.

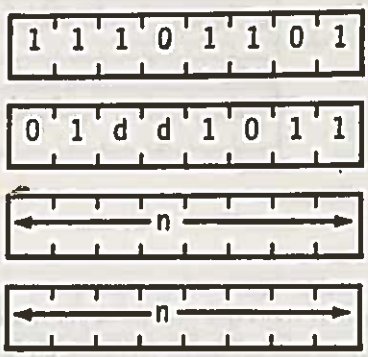


Ciclos: 5
Estados: 16
Indicadores: ninguno

LD dd, (nn)

dd_H←(nn+1), dd_L←(nn)

El contenido de la dirección nn se carga en la parte de orden bajo del par de registros dd y el contenido de la siguiente dirección de memoria más alta (nn+1) se carga en la parte alta de dd.

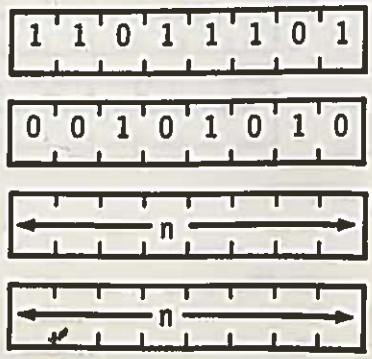


Ciclos: 6
Estados: 20
Indicadores: ninguno

LD IX, (nn)

IX_H←(nn+1), IX_L←(nn)

El contenido de la dirección nn se carga en la parte de orden bajo del registro índice IX y el contenido de la siguiente dirección de memoria más alta (nn+1) se carga en la parte de orden alto de IX.

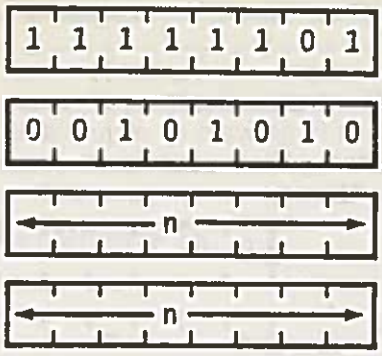


Ciclos: 6
Estados: 20
Indicadores: ninguno

LD IY (nn)

$IY_H \leftarrow (nn+1)$, $IY_L \leftarrow (nn)$

El contenido de la dirección nn se carga en la parte de orden bajo de registro índice IY y el contenido de la siguiente dirección de memoria más alta (nn+1) se carga en la parte de orden alto de IY.

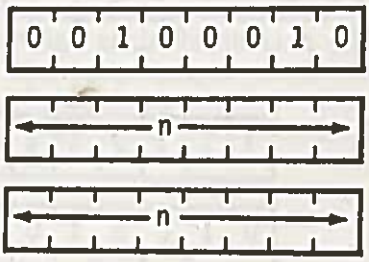


Ciclos: 6
Estados: 20
Indicadores: ninguno

LD (nn), HL

$(nn+1) \leftarrow H$, $(nn) \leftarrow L$

El contenido del registro L se carga en la dirección de memoria nn y el contenido del registro H se carga en la siguiente posición de dirección más alta nn+1.

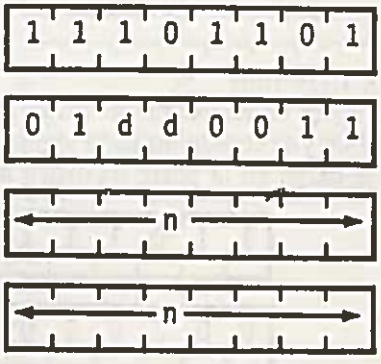


Ciclos: 5
Estados: 16
Indicadores: ninguno

LD (nn), dd

$(nn+1) \leftarrow dd_H$, $(nn) \leftarrow dd_L$

El octeto de orden bajo del par de registros dd se carga en la dirección de memoria nn; el octeto superior se carga en la dirección de memoria nn+1.

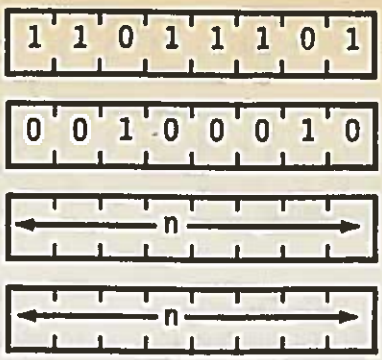


Ciclos: 6
Estados: 20
Indicadores: ninguno

LD (nn), IX

$(nn+1) \leftarrow IX_H$, $(nn) \leftarrow IX_L$

El octeto de orden bajo en el registro índice IX se carga en la dirección de memoria nn; el octeto de orden superior se carga en la siguiente dirección más alta nn+1.

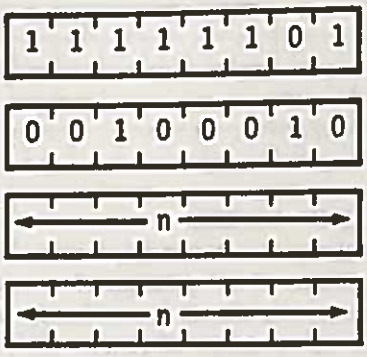


Ciclos: 6
Estados: 20
Indicadores: ninguno

LD (nn), IY

$(nn+1) \leftarrow IY_H$, $(nn) \leftarrow IY_L$

El octeto de orden bajo en el registro índice IY se carga en la dirección de memoria nn; el octeto de orden superior se carga en la posición de memoria nn+1.



Ciclos: 6
Estados: 20
Indicadores: ninguno

LD SP, HL

SP←HL

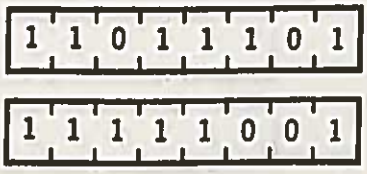
El contenido del par de registros HL se carga en el SP (puntero de pila).

Ciclos: 1
Estados: 6
Indicadores: ninguno

LD SP, IX

SP←IX

El contenido de 2 octetos del registro índice IX se carga en el SP (puntero de pila).



Ciclos: 2
Estados: 10
Indicadores: ninguno

LD SP, IY

SP←IY

El contenido de 2 octetos del registro índice IY se carga en el SP (puntero de pila).

Ciclos: 2
Estados: 10
Indicadores: ninguno

PUSH qq

$(SP-2) \leftarrow qq_L$, $(SP-1) \leftarrow qq_H$

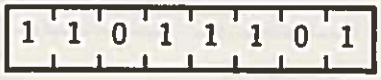
El contenido del par de registros qq se introduce en la pila LIFO (último en entrar, primero en salir) de memoria externa. El par de registros de puntero de pila (SP) retiene la dirección de 16 bits de la «parte superior» actual de la pila. Esta instrucción decrementa el SP y carga el octeto de orden alto del par de registros qq en la dirección de memoria ahora especificada por el SP; a continuación, vuelve a decrementar SP y carga el octeto de orden bajo de qq en la posición de memoria correspondiente a esta nueva dirección en el SP.

Ciclos: 3
Estados: 11
Indicadores: ninguno

PUSH IX

$(SP-2) \leftarrow IX_L$, $(SP-1) \leftarrow IX_H$

El contenido del registro índice IX se introduce en la pila. Esta instrucción decrementa primero el SP y carga el octeto de orden alto de IX en la dirección de memoria ahora especificada por el SP; entonces vuelve a decrementar el SP y carga el octeto de orden bajo en la posición de memoria correspondiente a esta nueva dirección en el SP.



Ciclos: 3
Estados: 15
Indicadores: ninguno

PUSH IY

$(SP-2) \leftarrow IY_L$, $(SP-1) \leftarrow IY_H$

El contenido del registro índice IY se introduce en la pila. Esta instrucción decrementa primero el SP y carga el octeto de orden alto de IY en la dirección de memoria ahora especificada por el SP; entonces vuelve a decrementar el SP y carga el octeto de orden bajo en la posición de memoria correspondiente a esta nueva dirección en el SP.

Ciclos: 4
Estados: 15
Indicadores: ninguno

POP qq

$qq_H \leftarrow (SP+1)$, $qq_L \leftarrow (SP)$

Los 2 octetos superiores de la pila se extraen y son introducidos inmediatamente en el par de registros qq. Esta instrucción carga primero

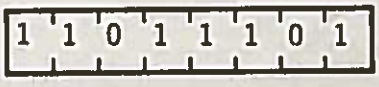
en la parte de orden bajo de qq el octeto en la posición de memoria correspondiente al contenido de SP; entonces el SP se incrementa y el contenido de la posición de memoria contigua correspondiente se carga en la parte de orden alto de qq y el SP se vuelve a incrementar ahora.

Ciclos: 3
Estados: 10
Indicadores: ninguno

POP IX

$IX_H \leftarrow (SP+1)$, $IX_L \leftarrow (SP)$

Los 2 octetos superiores de la pila se introducen en el registro índice IY. Esta instrucción carga primero en la parte de orden bajo de IY el octeto en la posición de memoria correspondiente al contenido de SP; entonces el contenido de SP; se incrementa el SP y el contenido de la posición de memoria contigua correspondiente se carga en la parte de orden alto de IX. El SP se vuelve a incrementar ahora.



Ciclos: 4
Estados: 14
Indicadores: ninguno

POP IY

$IY_H \leftarrow (SP+1)$, $IY_L \leftarrow (SP)$

Los 2 octetos superiores de la pila se introducen en el registro índice IY. Esta instrucción carga primero en la parte de orden bajo de IY el octeto en la posición de memoria correspondiente al contenido de SP; entonces el SP se incrementa y el contenido de la posición de memoria contigua correspondiente se carga en la parte de orden alto de IY. El SP vuelve a incrementarse ahora.

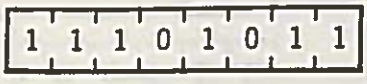
Ciclos: 4
Estados: 14
Indicadores: ninguno

INTERCAMBIO, TRANSFERENCIA DE BLOQUES Y GRUPO DE BUSQUEDA

EX DE, HL

$DE \leftrightarrow HL$

El contenido de 2 octetos de los pares de registros DE y HL son objeto de intercambio.



Ciclos: 1
Estados: 4
Indicadores: ninguno

EX AF, AF'

AF↔AF'

El contenido de 2 octetos de los pares de registros AF y AF' se intercambian.

Ciclos: 1
Estados: 4
Indicadores: ninguno

EXX

(BC)↔(BC'), (DE)↔(DE'), (HL)↔(HL')

Cada valor de 2 octetos en pares de registros BC, DE y HL se intercambia con el valor de 2 octetos en BC', DE' y HL', respectivamente.

Ciclos: 1
Estados: 4
Indicadores: ninguno

EX (SP), HL

H↔(SP+1), L↔(SP)

El octeto de orden bajo contenido en el par de registros HL se intercambia con el contenido de la dirección de memoria especificada por el contenido del par de registros SP y el octeto de orden alto de HL se intercambia con la siguiente dirección de memoria más alta (SP+1).

Ciclos: 5
Estados: 19
Indicadores: ninguno

EX (SP), IX

IX_H↔(SP+1), IX_L↔(SP)

El octeto de orden bajo en el registro índice IX se intercambia con el contenido de la dirección de memoria especificada por el contenido del par de registros SP y el octeto de orden alto de IX se intercambia con la siguiente dirección más alta (SP+1).

Ciclos: 6
Estados: 23
Indicadores: ninguno

EX (SP), IY

IY_H↔(SP+1), IY_L↔(SP)

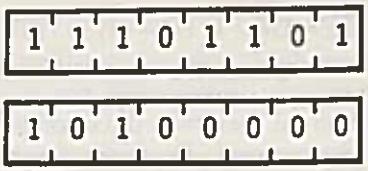
El octeto de orden bajo en el registro indice IY se intercambia con el contenido de la dirección de memoria especificada por el contenido del par de registros SP y el octeto de orden alto de IY se intercambia con la siguiente dirección de memoria más alta.

Ciclos: 6
Estados: 23
Indicadores: ninguno

LDI

(DE)←(HL), DE←DE+1, HL←HL+1, BC←BC−1

Un octeto de datos se transfiere desde la posición de memoria direccionada por el contenido del par de registros HL a la posición de memoria direccionada por el contenido del par de registros DE. A continuación, se incrementan ambos pares de registros y se decrementa el par de registros BC (contador de octetos).



Ciclos: 4
Estados: 16
Indicadores: H, N, P/V
H, N: poner a «0»
P/V: poner a «1» si BC−1≠0; poner a «0» en cualquier otro caso

LDIR

(DE)←(HL), DE←DE+1, HL←HL+1, BC←BC−1

Esta instrucción de 2 octetos transfiere un octeto de datos desde la posición de memoria direccionada por el contenido del par de registros HL a la posición de memoria direccionada por el par de registros DE. A continuación, se incrementan ambos pares de registros y se decrementa el par de registros (contador de octetos) BC. Si la acción de decremento hace que el BC pase a 0, se termina la instrucción. Si BC no es 0, el contador de programa se decrementa en 2 y se repite la instrucción. Observación: si BC se pone a 0 antes de la ejecución de la instrucción, la instrucción se iterará en bucle de 64 koctetos. Además, las interrupciones se identifican después de cada transferencia de datos.

Para BC≠0:

Ciclos: 5
Estados: 21

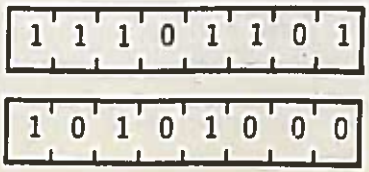
Para BC=0:

Ciclos: 4
Estados: 16
Indicadores: H, N, P/V: poner a «0»

LDD

(DE)←(HL), DE←DE−1, HL←HL−1, BC←BC−1

Esta instrucción de 2 octetos transfiere un octeto de datos desde la posición de memoria direccionada por el contenido del par de registros HL a la posición de memoria direccionada por el contenido del par de registros DE. A continuación, se decrementan ambos pares de registros incluyendo el par de registros (contadores de octetos) BC.



Ciclos: 4
Estados: 16
Indicadores: H, N, P/V
H, N: poner a «0»
P/V: poner a «1» si BC−1≠0; poner a «0» en cualquier otro caso

LDDR

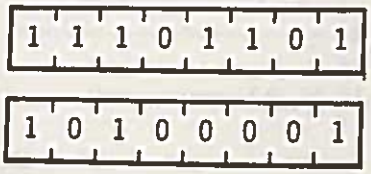
(DE)←(HL), DE←DE−1, HL←HL−1, BC←BC−1
Esta instrucción de 2 octetos transfiere un octeto de datos desde la posición de memoria direccionada por el contenido del par de registros HL a la posición de memoria direccionada por el contenido del par de registros DE. A continuación, se decrementan ambos registros, así como el contador de octetos, BC. Si la acción de decrementar hace que el BC pase a 0, se termina la instrucción. Si BC no es 0, el contador de programa se decrementa en 2 y se repite la instrucción. Observación: Si BC se pone a 0 antes de la ejecución de la instrucción, la instrucción se iterará en bucle de 64 koctetos. Además, las interrupciones se identificarán después de cada transferencia de datos.

Para BC≠0:
Ciclos: 5
Estados: 21

Para BC=0:
Ciclos: 4
Estados: 16
Indicadores: H, N, P/V: poner a «0»

CPI

A−(HL), HL←HL+1, BC←BC−1
El contenido de la posición de memoria direccionada por el par de registros HL se compara con el contenido del acumulador. En caso de una comparación verdadera, se pone a uno el bit de estado. A continuación, se incrementa HL y se decrementa el contador de octetos (par de registros BC).



Ciclos: 4
Estados: 16
Indicadores: S, Z, H, N, P/V
S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si A=(HL); poner a «0» en cualquier otro caso
H: poner a «1» si no hay acarreo negativo desde el bit 4; poner a «0» en cualquier otro caso
N: poner a uno
P/V: poner a «1» si BC−1≠0; poner a «0» en cualquier otro caso

CPIR

A−(HL), HL←HL+1, BC←BC−1

El contenido de la dirección de memoria direccionada por el registro HL se compara con el contenido del acumulador. En caso de una comparación verdadera, se pone a «1» un bit de estado. Se incrementa HL y se decrementa BC. Si se decrementa hace que BC pase a 0 o si A=(HL), se termina la instrucción. Si BC no es 0 y si A≠(HL), el contador de programa se decrementa en 2 y se repite la instrucción. Observación: Si BC se pone a «0» antes de la ejecución de la instrucción, la instrucción se iterará en bucle a través de 64 koctetos si no se encuentra ninguna coincidencia. Además, las interrupciones se identificarán después de cada comparación de datos.

Para BC≠0 y A≠(HL):

Ciclos: 5
Estados: 21

Para BC=0 o A=(HL):

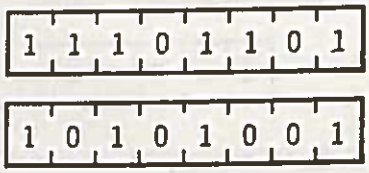
Ciclos: 4
Estados: 16
Indicadores: S, Z, H, N, P/V

S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si A=(HL); poner a «0» en cualquier otro caso
H: poner a «1» si no hay acarreo negativo desde el bit 4; poner a «0» en cualquier otro caso
N: poner a «1»
P/V: poner a «1» si BC−1≠0; poner a «0» en cualquier otro caso

CPD

A−(HL), HL←HL−1, BC←BC−1

El contenido de la posición de memoria direccionada por el par de registros HL se compara con el contenido del acumulador. En caso de una comparación verdadera, se pone a «1» un bit de estado. Los pares HL y BC se decrementan.



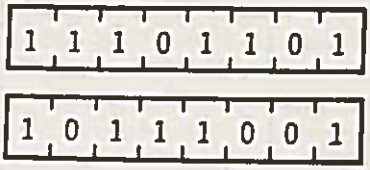
Ciclos: 4
Estados: 16
Indicadores: S, Z, H, N, P/V

S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si A=(HL); poner a «0» en cualquier otro caso
H: poner a «1» si no hay acarreo negativo desde el bit 4; poner a «0» en cualquier otro caso
N: poner a «1»
P/V: poner a «1» si BC−1≠0; poner a «0» en cualquier otro caso

CPDR

A−(HL), HL←HL−1, BC←BC−1

El contenido de la posición de memoria direccionada por el par de registros HL se compara con el contenido del acumulador. En caso de una comparación verdadera, se pone a «1» un bit de estado. Los pares de registro HL y BC se decrementan. Si se decrementa hace que BC pase a «0» o si A=(HL), se termina la instrucción. Si BC no es «0» y A≠(HL), el contador de programa se decrementa en 2 y se repite la instrucción. Observación: Si BC se pone a «0» antes de la ejecución de la instrucción, la instrucción se iterará en bucle a través de 64 koctetos si no se encuentra ninguna coincidencia. Además, las interrupciones se identificarán después de cada comparación de datos.



Para BC≠0 y A≠(HL):

Ciclos: 5
Estados: 21

Para BC=0 o A=(HL):

Ciclos: 4
Estados: 16
Indicadores: S, Z, H, N, P/V

S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si A=(HL); poner a «0» en cualquier otro caso
H: poner a «1» si no hay acarreo negativo desde el bit 4; poner a «0» en cualquier otro caso
N: poner a «1»
P/V: poner a «1» si BC−1≠0; poner a «0» en cualquier otro caso

GRUPO LOGICO Y ARITMETICO DE OCHO BITS

ADD A, r

A←A+r

El contenido del registro r se añade al contenido del acumulador y el resultado se almacena en el acumulador.

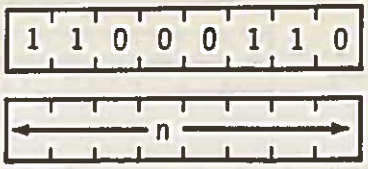
Ciclos: 1
Estados: 4
Indicadores: S, Z, H, N, C, P/V

S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso
H: poner a «1» si hay acarreo desde el bit 3; poner a «0» en cualquier otro caso
N: poner a «0»
C: poner a «1» si hay acarreo desde el bit 7; poner a «0» en cualquier otro caso
P/V: poner a «1» si hay desbordamiento; poner a «0», en cualquier otro caso

ADD A, n

A←A+n

El entero n se añade al contenido del acumulador y los resultados se almacenan en el acumulador.



Ciclos: 2
Estados: 7
Indicadores: S, Z, H, N, C, P/V

S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso
H: poner a «1» si hay acarreo desde el bit 3; poner a «0» en cualquier otro caso
N: poner a «0»
C: poner a «1» si hay acarreo desde el bit 7; poner a «0» en cualquier otro caso
P/V: poner a «1» si hay desbordamiento; poner a «0» en cualquier otro caso

ADD A, (HL)

A←A+(HL)

El octeto en la dirección de memoria especificada por el contenido del par de registros HL se añade al contenido del acumulador y el resultado se almacena en el acumulador.

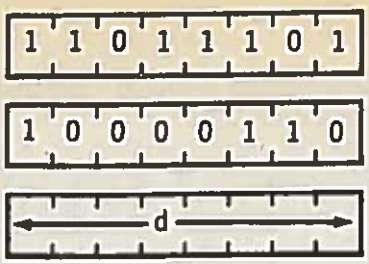
Ciclos: 2
Estados: 7
Indicadores: S, Z, H, N, C, P/V

S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso
H: poner a «1» si hay acarreo desde el bit 3; poner a «0» en cualquier otro caso
N: poner a «0»
C: poner a «1» si hay acarreo desde el bit 7; poner a «0» en cualquier otro caso
P/V: poner a «1» si hay desbordamiento; poner a «0» en cualquier otro caso

ADD A, (IX+d)

A←A+(IX+d)

El contenido del registro índice IX se añade a un desplazamiento d para apuntar a una dirección en memoria. El contenido de esta dirección se añade entonces al contenido del acumulador y el resultado se almacena en el acumulador.



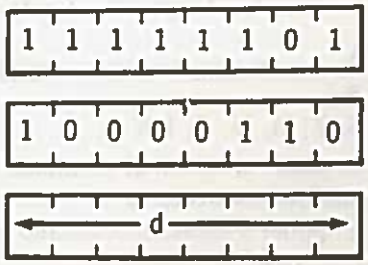
Ciclos: 5
Estados: 19
Indicadores: S, Z, H, N, C, P/V

S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso
H: poner a «1» si hay acarreo desde el bit 3; poner a «0» en cualquier otro caso
N: poner a «0»
C: poner a «1» si hay acarreo desde bit 7; poner a «0» en cualquier otro caso
P/V: poner a «1» si hay desbordamiento; poner a «0» en cualquier otro caso

ADD A, (IY+d)

A←A+(IY+d)

El contenido del registro índice IY se añade a un desplazamiento d para apuntar a una dirección en memoria. El contenido de esta dirección se añade entonces al contenido del acumulador y el resultado se almacena en el acumulador.



Ciclos: 5
Estados: 19
Indicadores: S, Z, H, N, C, P/V

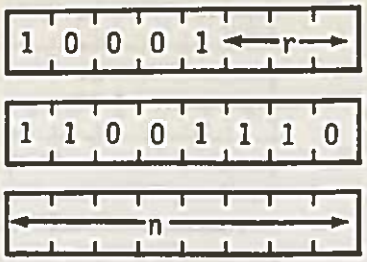
S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso
H: poner a «1» si hay acarreo desde el bit 3; poner a «0» en cualquier otro caso
N: poner a «1»
C: poner a «1» si hay acarreo desde el bit 7; poner a «0» en cualquier otro caso
P/V: poner a «1» si hay desbordamiento; poner a «0» en cualquier otro caso

ADC A, s

A←A+s+CY

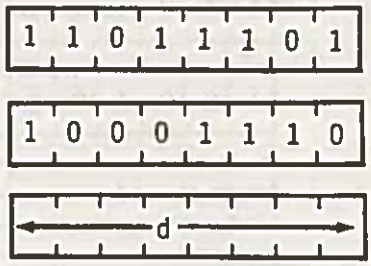
El operando s es cualquiera de r, n, (HL), (IX+d) o (IY+1) según se define para la instrucción ADD análoga. Estas diversas combinaciones posibles de operandos de código operación se ensamblan en el código objeto como sigue:

ADC A, r
ADC A, n

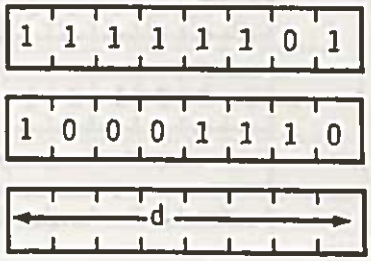


ADC A, (HL)

ADC A, (IX+d)



ADC A, (IY+d)



El operando s, junto con el indicador de acarreo («C» en el registro F), se añade al contenido del acumulador y el resultado se almacena en el acumulador.

Instrucción	*Ciclos*	*Estados*
ADC A, r	1	4
ADC A, n	2	7
ADC A, (HL)	2	7
ADC A, (IX+d)	5	19
ADC A, (IY+d)	5	19

Indicadores: S, Z, H, N, C, P/V

S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso
H: poner a «1» si hay acarreo desde el bit 3; poner a «0» en cualquier otro caso
N: poner a «0»
C: poner a «1» si hay acarreo desde el bit 7; poner a «0» en cualquier otro caso
P/V: poner a «1» si hay desbordamiento; poner a «0» en cualquier otro caso

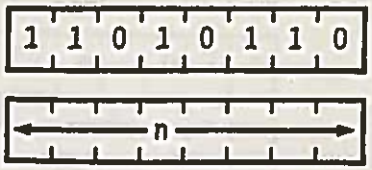
SUB s

A←A−s

El operando s se resta del contenido del acumulador y el resultado se almacena en el acumulador.

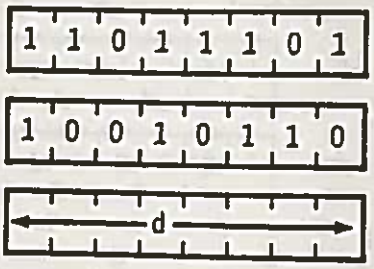
SUB r

SUB n

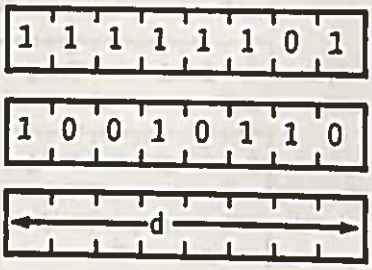


SUB (HL)

SUB (IX+d)



SUB (IY+d)



Instrucción	*Ciclos*	*Estados*
SUB r	1	4
SUB n	2	7
SUB (HL)	2	7
SUB (IX+d)	5	19
SUB (IY+d)	5	19

Indicadores: S, Z, H, N, C, P/V

S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso

Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso

H: poner a «1» si no hay acarreo negativo desde el bit 4; poner a «0» en cualquier otro caso

N: poner a «1»

C: poner a «1» si no hay acarreo negativo; poner a «0» en cualquier otro caso

P/V: poner a «1» si hay desbordamiento; poner a «0» en cualquier otro caso

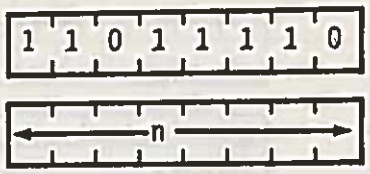
SBC A, s

A←A−s−CY

El operando s, junto con el indicador de acarreo («C» en el registro F), se resta del contenido del acumulador y el resultado se almacena en el acumulador.

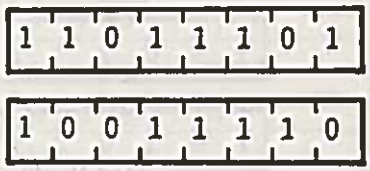
SBC A, r

SBC A, n

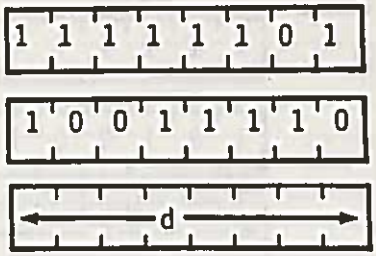


SBC A, (HL)

SBC A, (IX+d)



SBC A, (IY+d)



Instrucción	*Ciclos*	*Estados*
SBC A, r	1	4
SBC A, n	2	7
SBC A, (HL)	2	7
SBC A, (IX+d)	5	19
SBC A, (IY+d)	5	19

Indicadores: S, Z, H, N, C, P/V

S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso

Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso

H: poner a «1» si no hay acarreo negativo desde el bit 4; poner a «0» en cualquier otro caso

N: poner a «1»

C: poner a «1» si no hay acarreo negativo; poner a «0» en cualquier otro caso

P/V: poner a «1» si hay desbordamiento; poner a «0» en cualquier otro caso

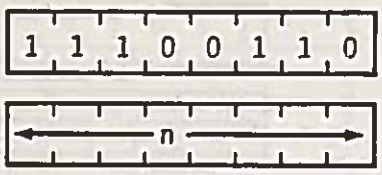
AND s

$A \leftarrow A \wedge s$

Una operación AND lógica, bit por bit, se realiza entre el octeto especificado por el operando s y el octeto contenido en el acumulador; el resultado se almacena en el acumulador.

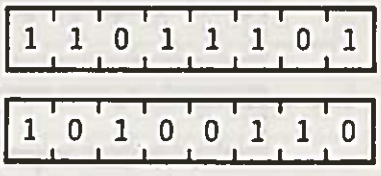
AND r

AND n

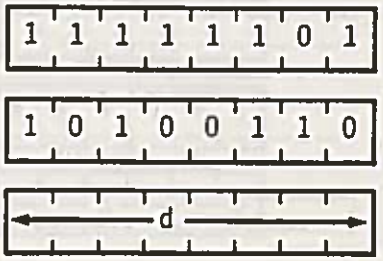


AND (HL)

AND (IX+d)



AND (IY+d)



Instrucción	*Ciclos*	*Estados*
AND r	1	4
AND n	2	7
AND (HL)	2	7
AND (IX+d)	5	19
AND (IY+d)	5	19

Indicadores: S, Z, H, N, C, P/V

S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso

Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso

H: poner a «1»

N: poner a «0»

C: poner a «0»

P/V: poner a «1» si la paridad es par; poner a «0» en cualquier otro caso

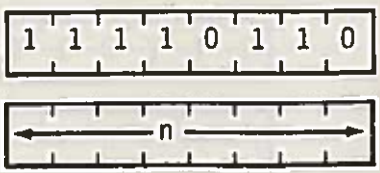
OR s

A←A ∨ s

Una operación OR lógica, bit por bit, se realiza entre el octeto especificado por el operando s y el octeto contenido en el acumulador; el resultado se almacena en el acumulador.

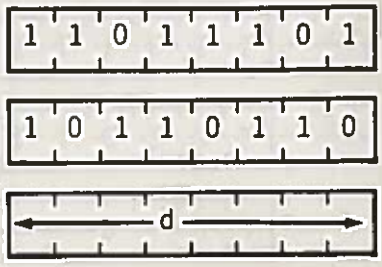
OR r

OR n

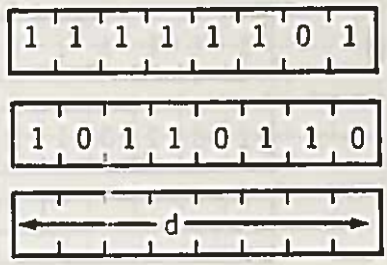


OR (HL)

OR (IX+d)



OR (IY+d)



Instrucción	*Ciclos*	*Estados*
OR r	1	4
OR n	2	7
OR (HL)	2	7
OR (IX+d)	5	19
OR (IY+d)	5	19

Indicadores: S, Z, H, N, C, P/V

S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso

Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso

H: poner a «1»

N: poner a «0»

C: poner a «0»

P/V: poner a «1» si la paridad es par; poner a «0» en cualquier otro caso

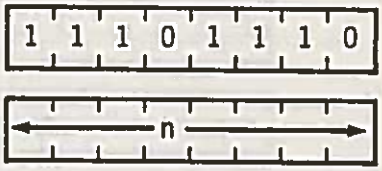
XOR s

$A \leftarrow A \oplus s$

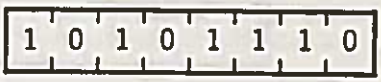
Una operación OR exclusiva, bit por bit, se realiza entre el octeto especificado por el operando s y el octeto contenido en el acumulador; el resultado se almacena en el acumulador.

XOR r

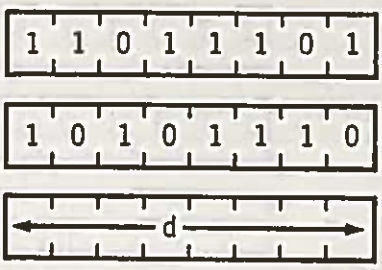
XOR n



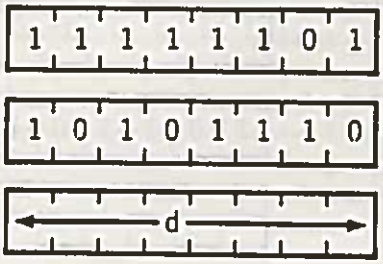
XOR (HL)



XOR (IX+d)



XOR (IY+d)



Instrucción	*Ciclos*	*Estados*
XOR r	1	4
XOR n	2	7
XOR (HL)	2	7
XOR (IX+d)	5	19
XOR (IY+d)	5	19

Indicadores: S, Z, H, N, C, P/V

S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso

Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso

H: poner a «1»

N: poner a «0»

C: poner a «0»

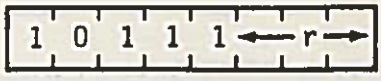
P/V: poner a «1» si la paridad es par; poner a «0» en cualquier otro caso

CP s

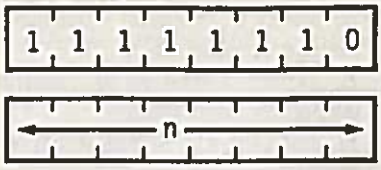
A−s

El contenido del operando s se compara con el contenido del acumulador. Si hay una comparación verdadera, se pone a «1» un indicador.

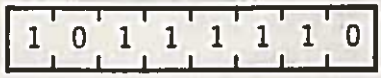
CP r



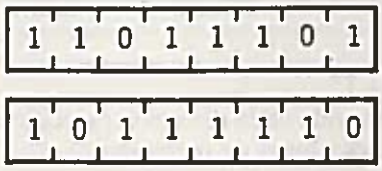
CP n



CP (HL)



CP (IX+d)



CP (IY+d)

Instrucción	*Ciclos*	*Estados*
CP r	1	4
CP n	2	7
CP (HL)	2	7
CP (IX+d)	5	19
CP (IY+d)	5	19

Indicadores: S, Z, H, N, C, P/V

S: poner a «1» si el resultado es necesario; poner a «0» en cualquier otro caso

Z: poner a «1» si el resultado es «0»; poner a «0» en cualquier otro caso

H: poner a «1» si no hay acarreo negativo desde el bit 4; poner a «0» en cualquier otro caso

N: poner a «1»

C: poner a «1» si no hay acarreo negativo; poner a «0» en cualquier otro caso

P/V: poner a «1» si hay desbordamiento; poner a «0» en cualquier otro caso

INC r

r←r+1

Se incrementa el registro r.

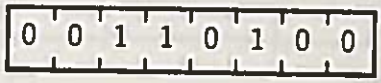
Ciclos: 1
Estados: 4
Indicadores: S, Z, H, N, P/V

S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si el resultado es «0»; poner a «0» en cualquier otro caso
H: poner a «1» si hay acarreo desde el bit 3; poner a «0» en cualquier otro caso
N: poner a «0»
P/V: poner a «1» si r era 7FH antes de la operación; poner a «0» en cualquier otro caso.

INC (HL)

(HL)←(HL)+1

El octeto contenido en la dirección especificada por el contenido del par de registros HL es objeto de incremento.



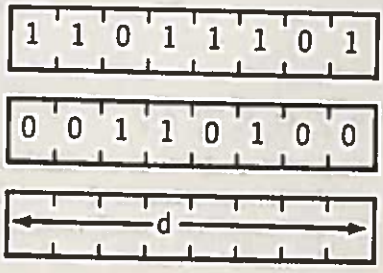
Ciclos: 3
Estados: 11
Indicadores: S, Z, H, N, P/V

S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso
H: poner a «1» si hay acarreo desde el bit 3; poner a «0» en cualquier otro caso
N: poner a «0»
P/V: poner a «1» si (HL) era 7FH antes de la operación; poner a «0» en cualquier otro caso

INC (IX+d)

(IX+d)←(IX+d)+1

El contenido del registro índice IX se añade a un entero de desplazamiento d, de complemento a 2, para apuntar a una dirección de memoria. A continuación, se incrementa el contenido de esta dirección.



Ciclos: 6
Estados: 23
Indicadores: S, Z, H, N, P/V

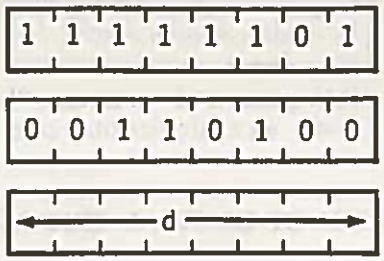
S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso

H: poner a «1» si hay acarreo desde el bit 3; poner a «0» en cualquier otro caso
N: poner a «0»
P/V: poner a «1» si (IX+d) era 7FH antes de la operación; poner a «0» en cualquier otro caso

INC (IY+d)

(IY+d)←(IY+d)+1

El contenido del registro índice IY se añade a un entero de desplazamiento d, de complemento a 2, para apuntar a una dirección en memoria. El contenido de esta dirección se incrementa a continuación.



Ciclos: 6
Estados: 23
Indicadores: S, Z, H, N, P/V

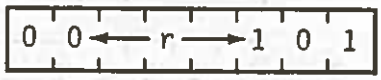
S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso
H: poner a «1» si hay acarreo desde el bit 3; poner a «0» en cualquier otro caso
N: poner a «0»
P/V: poner a «1» si (IY+d) era 7FH antes de la operación; poner a «0» en cualquier otro caso

DEC m

m←m−1

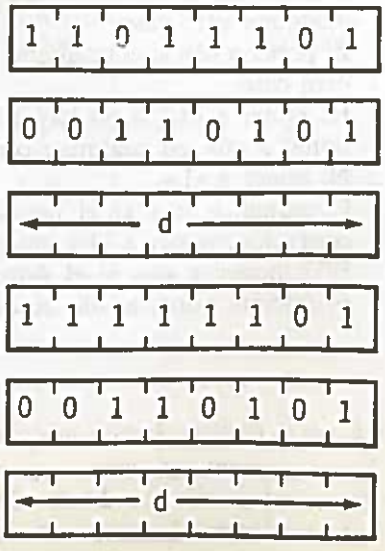
El octeto especificado por el operando m es objeto decremento.

DEC r



DEC (HL)

DEC (IX+d)



DEC (IY+d)

Instrucción	*Ciclos*	*Estados*
DEC r	1	4
DEC (HL)	3	11
DEC (IX+d)	6	23
DEC (IY+d)	6	23

Indicadores: S, Z, H, N, P/V

S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso
H: poner a «1» si no hay acarreo desde el bit 4; poner a «0» en cualquier otro caso
N: poner a «1»
P/V: poner a «1» si m era 80H antes de la operación; poner a «0» en cualquier otro caso

GRUPOS DE CONTROL DE LA CPU Y ARITMETICOS DE USO GENERAL

CPL

$A \leftarrow \overline{A}$

El contenido del acumulador se invierte (complemento a «1»).

Ciclos: 1
Estados: 4

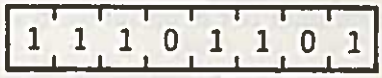
Indicadores: H, N

H: poner a «1»
N: poner a «1»

NEG

$A \rightarrow 0 - A$

El contenido del acumulador se niega (complemento a 2). Es lo mismo que restar el contenido del acumulador de 0.



Ciclos: 2
Estados: 8

Indicadores: S, Z, H, N, C, P/V

S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso
H: poner a «1» si no hay acarreo negativo desde el bit 4; poner a «0» en cualquier otro caso
N: poner a «1»
C: poner a «1» si el acumulador era 00H antes de la operación; poner a «0» en cualquier otro caso
P/V: poner a «1» si el acumulador era 80H antes de la operación; poner a «0» en cualquier otro caso

CCF

$CY \leftarrow \overline{CY}$

El indicador C en el registro F está invertido.

Ciclos: 1
Estados: 4
Indicadores: H, N, C

H: se copiará acarreo anterior
N: poner a «0»
C: poner a «1» si CY era 0 antes de la operación; poner a «0» en cualquier otro caso.

SCF

CY←1
El indicador C en el registro F se pone a «1».

Ciclos: 1
Estados: 4
Indicadores: H, N, C

H: poner a «0»
N: poner a «0»
C: poner a «1»

NOP

– – – – –

El procesador central no realiza ninguna operación durante este ciclo de máquina.

Ciclos: 1
Estados: 4
Indicadores: ninguno

DAA

– – – – –

La instrucción ajusta condicionalmente el acumulador para las operaciones de suma y resta en BCD. Para la suma (ADD, ADC, INC) o resta (SUB, SBC, DEC, NEG), la siguiente tabla indica la operación realizada:

OPERACION	C ANTES DAA	VALOR HEXA-DECI-MAL EN DIGITO SUPERIOR (bit 7-4)	H ANTES DAA	VALOR HEXA-DECI-MAL EN DIGITO SUPERIOR (bit 3-0)	NUMERO AÑADIDO A OCTETO	C DESPUES DAA
ADD ADC INC	0	0-9	0	0-9	00	0
	0	0-8	0	A-F	06	0
	0	0-9	1	0-3	06	0
	0	A-F	0	0-9	60	1
	0	9-F	0	A-F	66	1
	0	A-F	1	0-3	66	1
	1	0-2	0	0-9	60	1
	1	0-2	0	A-F	66	1
	1	0-3	1	0-3	66	1
SUB SBC DEC NEG	0	0-9	0	0-9	00	0
	0	0-8	1	6-F	FA	0
	1	7-F	0	0-9	A0	1
	1	6-F	1	6-F	9A	1

M CICLOS: 1 T ESTADOS: 4 4 MHZ E.T.: 1,00

Ciclos: 1
Estados: 4
Indicadores: S, Z, H, C, P/V

S: poner a «1» si el bit más significativo del acumulador es «1» después de la operación; poner a «0» en cualquier otro caso
Z: poner a «1» si el acumulador es «0» después de la operación; poner a «0» en cualquier otro caso
H: ver instrucción
C: ver instrucción
P/V: poner a «1» si el acumulador es de paridad par después de la operación; poner a «0» en cualquier otro caso

HALT

La instrucción HALT suspende la operación del procesador central hasta que se reciba una subsiguiente interrupción o puesta a «0». Mientras se está en el estado de interrupción, el procesador ejecutará NOPs para mantener la lógica de regeneración de memoria.

Ciclos: 1
Estados: 4
Indicadores: ninguno

DI

IFF←0
DI inhibe la interrupción enmascarable poniendo a «0» los flip-flops de la habilitación de interrupción (IFF1 y IFF2). Observación: Esta instrucción inhibe la interrupción enmascarable durante su ejecución.

Ciclos: 1
Estados: 4
Indicadores: ninguno

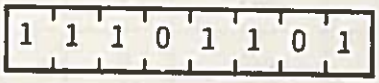
EI

IFF←1
EI desbloquea la interrupción enmascarable poniendo a «1» los flip-flops de habilitación de interrupción (IFF1 y IFF2). Observación: Esta instrucción inhibe la interrupción enmascarable durante su ejecución.

Ciclos: 1
Estados: 4
Indicadores: ninguno

IM 0

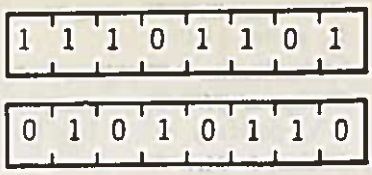
La instrucción IM 0 activa el modo de interrupción 0. En este modo, el dispositivo de interrupción puede insertar cualquier instrucción en el bus de datos y permitir que el procesador central la ejecute.



Ciclos: 2
Estados: 8
Indicadores: ninguno

IM 1

La instrucción IM 1 activa el modo de interrupción 1. En este modo, el procesador responderá a una interrupción ejecutando un reinicio de la posición 0038H.



Ciclos: 2
Estados: 8
Indicadores: ninguno

IM 2

La instrucción IM 2 activa el modo de interrupción 2. Este modo permite una llamada indirecta a cualquier posición en memoria. Con este modo, el procesador central forma una dirección de memoria de 16 bits. Los 8 bits superiores son el contenido del registro de vectores de interrupción I y los 8 bits inferiores se suministran por el dispositivo de interrupción.

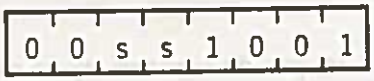
Ciclos: 2
Estados: 8
Indicadores: ninguno

GRUPO ARITMETICO DE 16 BITS

ADD HL, ss

HL←HL+ss

El contenido del par de registros ss se añade al contenido del par de registros HL y el resultado se almacena en HL.



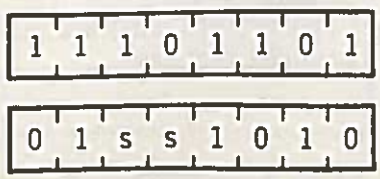
Ciclos: 3
Estados: 11
Indicadores: H, N, C

H: poner a «1» si hay acarreo fuera del bit 11; poner a «0» en cualquier otro caso
H: poner a «0»
C: poner a «1» si hay acarreo desde el bit 15; poner a «0» en cualquier otro caso

ADD HL, ss

HL←HL+ss+CY

El contenido del par de registros ss se suma con el indicador de acarreo al contenido del par de registros HL y el resultado se almacena en HL.



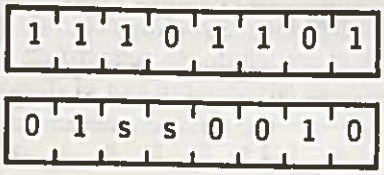
Ciclos: 4
Estados: 15
Indicadores: S, Z, H, N, C, P/V

S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso
H: poner a «1» si hay acarreo fuera del bit 11; poner a «0» en cualquier otro caso
N: poner a «0»
C: poner a «1» si hay acarreo desde el bit 15; poner a «0» en cualquier otro caso
P/V: poner a «1» si hay desbordamiento; poner a «0» en cualquier otro caso

SBC HL, ss

HL←HL−ss−CY

El contenido del par de registros ss y el indicador de acarreo se restan del contenido del par de registros HL y el resultado se almacena en HL.



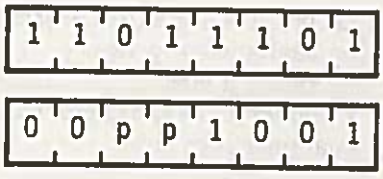
Ciclos: 4
Estados: 15
Indicadores: S, Z, H, N, C, P/V

S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso
H: poner a «1» si no hay acarreo negativo desde el bit 12; poner a «0» en cualquier otro caso
N: poner a «1»
C: poner a «1» si no hay acarreo negativo; poner a «0» en cualquier otro caso
P/V: poner a «1» si hay desbordamiento; poner a «0» en cualquier otro caso

ADD IX, pp

IX←IX+pp

El contenido del par de registros pp se suma al contenido del registro índice IX y los resultados se almacenan en IX.



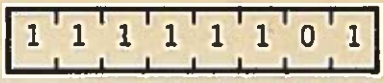
Ciclos: 4
Estados: 15
Indicadores: H, N, C

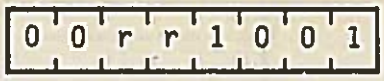
H: poner a «1» si hay acarreo fuera del bit 11; poner a «0» en cualquier otro caso
N: poner a «0»
C: poner a «1» si hay acarreo desde el bit 15; poner a «0» en cualquier otro caso

ADD IY, rr

IY←IY+rr

El contenido del par de registros rr se suma al contenido del registro índice IY y el resultado se almacena en IY.





Ciclos: 4
Estados: 15
Indicadores: H, N, C

H: poner a «1» si hay acarreo fuera del bit 11; poner a «0» en cualquier otro caso
N: poner a «0»
C: poner a «1» si hay acarreo desde el bit 15; poner a «0» en cualquier otro caso

INC ss

ss←ss+1

Se incrementa el contenido del par de registros ss.

Ciclos: 1
Estados: 6
Indicadores: ninguno

INC IX

IX←IX+1

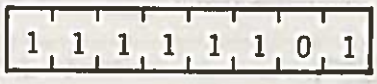
Se incrementa el contenido del registro índice IX.

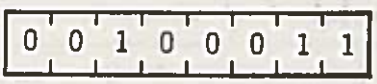
Ciclos: 2
Estados: 10
Indicadores: ninguno

INC IY

IY←IY+1

Se incrementa el contenido del registro índice IY.



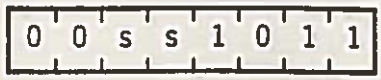


Ciclos: 2
Estados: 10
Indicadores: ninguno

DEC ss

ss←ss−1

Se decrementa el contenido del par de registros ss.

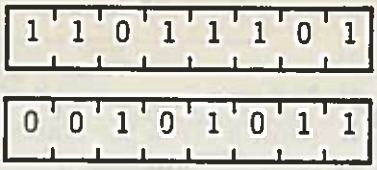


Ciclos: 1
Estados: 6
Indicadores: ninguno

DEC IX

IX←IX−1

Se decrementa el contenido del registro índice IX.

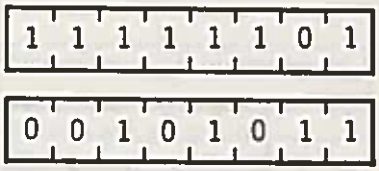


Ciclos: 2
Estados: 10
Indicadores: ninguno

DEC IY

IY←IY−1

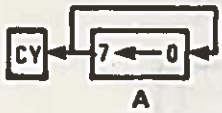
Se decrementa el contenido del registro índice IY.



Ciclos: 2
Estados: 10
Indicadores: ninguno

GRUPO DE ROTACION Y DESPLAZAMIENTO

RLCA

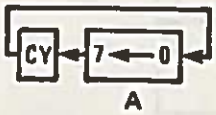


El contenido del acumulador se gira a la izquierda. El contenido del bit 7 se copia en el indicador de acarreo y también en el bit 0.

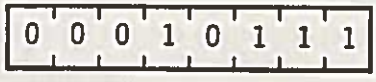
Ciclos: 1
Estados: 4
Indicadores: H, N, C

H: poner a «0»
N: poner a «0»
C: datos desde el bit 7 del acumulador

RLA

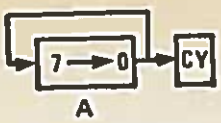


El contenido del acumulador se gira a la izquierda. El contenido del bit 7 se copia en el indicador del acarreo y el contenido anterior de este indicador se copia en el bit 0.

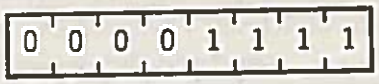


Ciclos: 1
Estados: 4
Indicadores: H, N, C

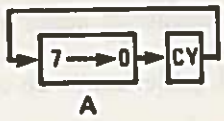
H: poner a «0»
N: poner a «0»
C: datos desde el bit 7 del acumulador

RRCA 

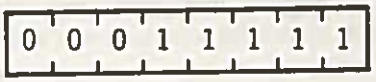
El contenido del acumulador se gira a la derecha. El contenido del bit 0 se copia en el bit 7 y también en el indicador del acarreo.



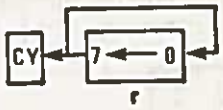
Ciclos: 1
Estados: 4
Indicadores: H, N, C
H: poner a «0»
N: poner a «0»
C: datos desde el bit 0 del acumulador

RRA 

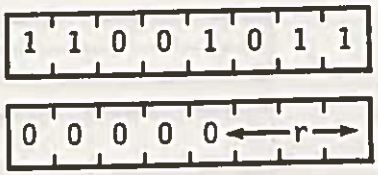
El contenido del acumulador se gira a la derecha. El contenido del bit 0 se copia en el indicador de acarreo y el contenido anterior del indicador de acarreo se copia en el bit 7.



Ciclos: 1
Estados: 4
Indicadores: H, N, C
H: poner a «0»
N: poner a «0»
C: datos desde el bit 0 del acumulador

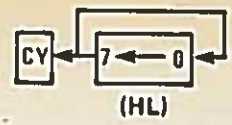
RLC r 

El contenido de 8 bits del registro r se gira a la izquierda. El contenido del bit 7 se copia en el indicador de acarreo y también en el bit 0.

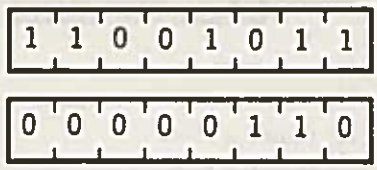


Ciclos: 2
Estados: 8
Indicadores: S, Z, H, N, C, P/V
S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso
H: poner a «0»
N: poner a «0»
C: datos desde bit 7 de registro fuente
P/V: poner a «1» si la paridad es par; poner a «0» en cualquier otro caso

RLC (HL)



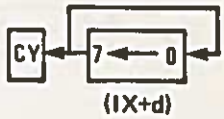
El contenido de la dirección de memoria, especificada por el contenido del par de registros HL, se gira a la izquierda. El contenido del bit 7 se copia en el indicador de acarreo y también en el bit 0.



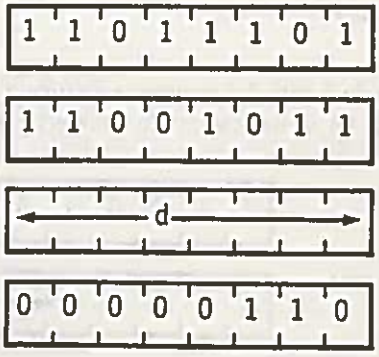
Ciclos: 4
Estados: 15
Indicadores: S, Z, H, N, C, P/V

S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso
H: poner a «0»
N: poner a «0»
C: datos desde bit 7 de registro fuente
P/V: poner a «1» si la paridad es par; poner a «0» en cualquier otro caso

RLC (IX+d)



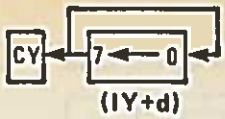
El contenido de la dirección de memoria, especificada por la suma del contenido del registro índice IX y un entero de desplazamiento d, de complemento a 2, se gira a la izquierda. El contenido del bit 7 se copia en el indicador de acarreo y también en el bit 0.



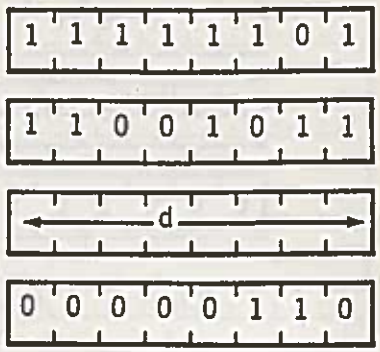
Ciclos: 6
Estados: 23
Indicadores: S, Z, H, N, C, P/V

S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso
H: poner a «0»
N: poner a «0»
C: datos desde bit 7 de registro fuente
P/V: poner a «1» si la paridad es par; poner a «0» en cualquier otro caso

RLC (IY+d)



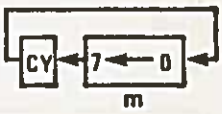
El contenido de la dirección de memoria, especificada por la suma del contenido del registro índice IY y un entero de desplazamiento d, de complemento a 2, se gira a la izquierda. El contenido del bit 7 se copia en el indicador de acarreo y también en el bit 0.



Ciclos: 6
Estados: 23
Indicadores: S, Z, H, N, C, P/V

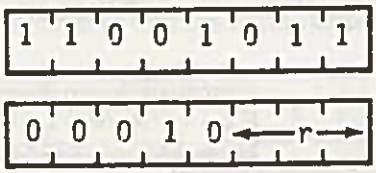
S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso
H: poner a «0»
N: poner a «0»
C: datos desde bit 7 de registro fuente
P/V: poner a «1» si la paridad es par; poner a «0» en cualquier otro caso

RL m

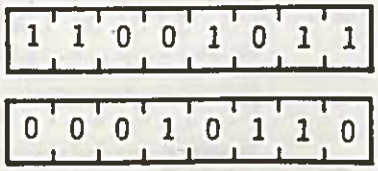


El contenido del operando m se gira a la izquierda. El contenido del bit 7 se copia en el indicador de acarreo y el contenido anterior de este indicador se copia en el bit 0.

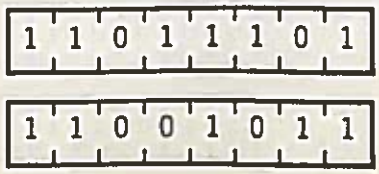
RL r

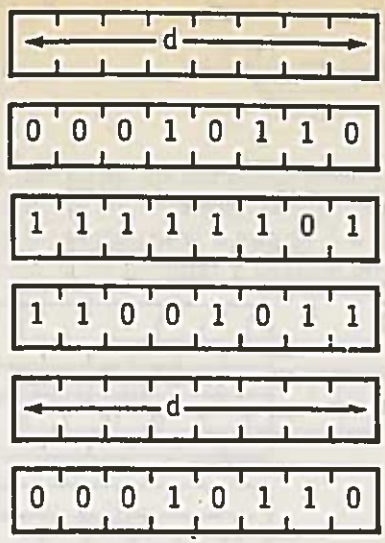


RL (HL)



RL (IX+d)





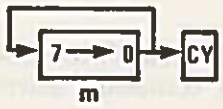
RL (IY+d)

Instrucción	*Ciclos*	*Estados*
RL r	2	8
RL (HL)	4	15
RL (IX+d)	6	23
RL (IY+d)	6	23

Indicadores: S, Z, H, N, C, P/V

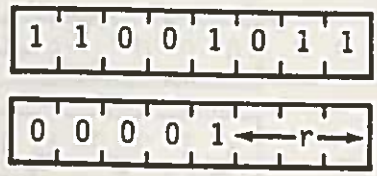
S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso
H: poner a «0»
N: poner a «0»
C: datos desde bit 7 de registro fuente
P/V: poner a «1» si la paridad es par; poner a «0» en cualquier otro caso

RRC m

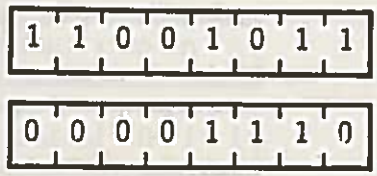


El contenido del operando m se gira a la derecha. El contenido del bit 0 se copia en el indicador de acarreo y también en el bit 7.

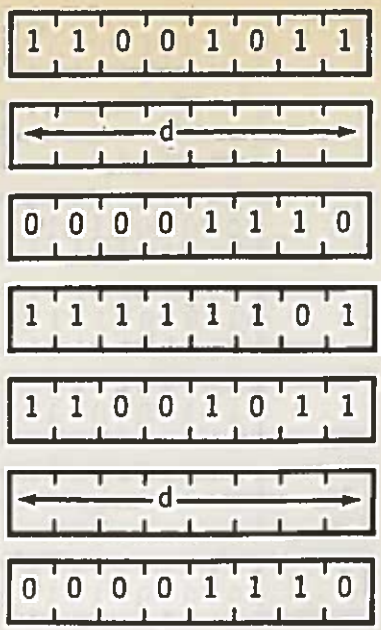
RRC r



RRC (HL)



RRC (IX+d)



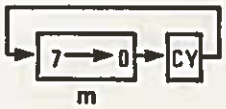
RRC (IY+d)

Instrucción	*Ciclos*	*Estados*
RRC r	2	8
RRC (HL)	4	15
RRC (IX+d)	6	23
RRC (IY+d)	6	23

Indicadores: S, Z, H, N, C, P/V

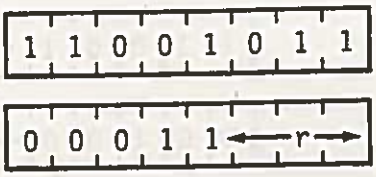
S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso
H: poner a «0»
N: poner a «0»
C: datos desde bit 0 de registro fuente
P/V: poner a «1» si la paridad es par; poner a «0» en cualquier otro caso

RR m

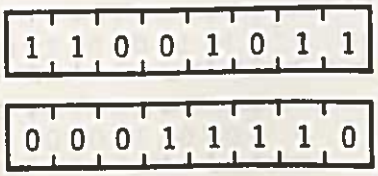


El contenido del operando m se gira a la derecha. El contenido del bit 0 se copia en el indicador de acarreo y el contenido anterior de este último se copia en el bit 7.

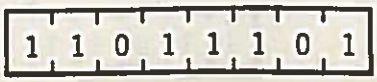
RR r

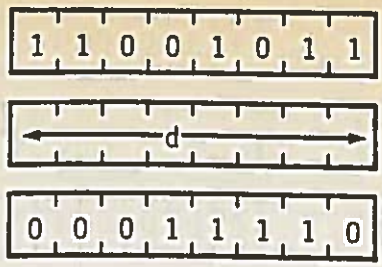


RR (HL)

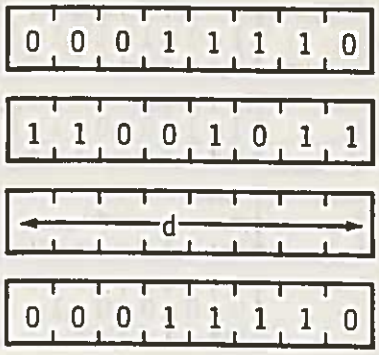


RR (IX+d)





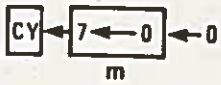
RR (IY+d)



Instrucción	*Ciclos*	*Estados*
RR r	2	8
RR (HL)	4	15
RR (IX+d)	6	23
RR (IY+d)	6	23

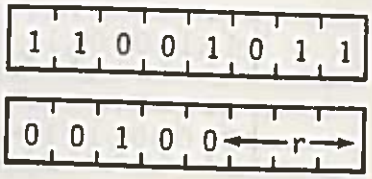
Indicadores: S, Z, H, N, C, P/V

S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso
H: poner a «0»
N: poner a «0»
C: datos desde bit 0 de registro fuente
P/V: poner a «1» si la paridad es par; poner a «0» en cualquier otro caso

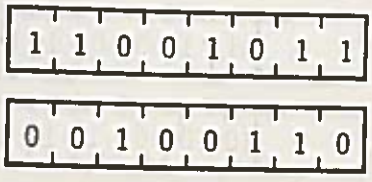
SLA m 

Un desplazamiento aritmético a la izquierda se realiza en el contenido del operando m. Se poner a «0» el bit 0. El contenido del bit 7 se copia en el indicador de acarreo.

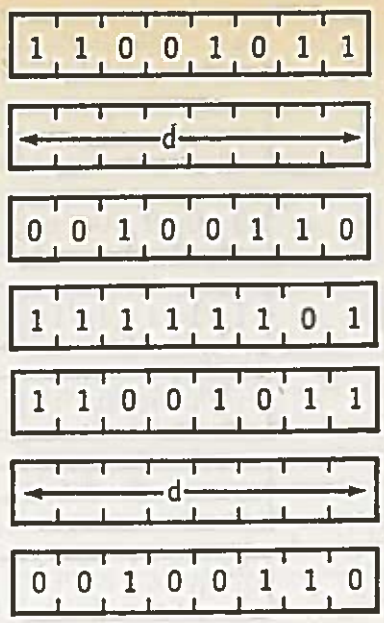
SLA r



SLA (HL)



SLA (IX+d)

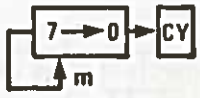


Instrucción	*Ciclos*	*Estados*
SLA r	2	8
SLA (HL)	4	15
SLA (IX+d)	6	23
SLA (IY+d)	6	23

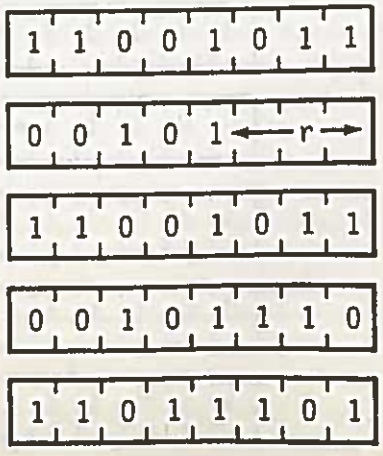
Indicadores: S, Z, H, N, C, P/V

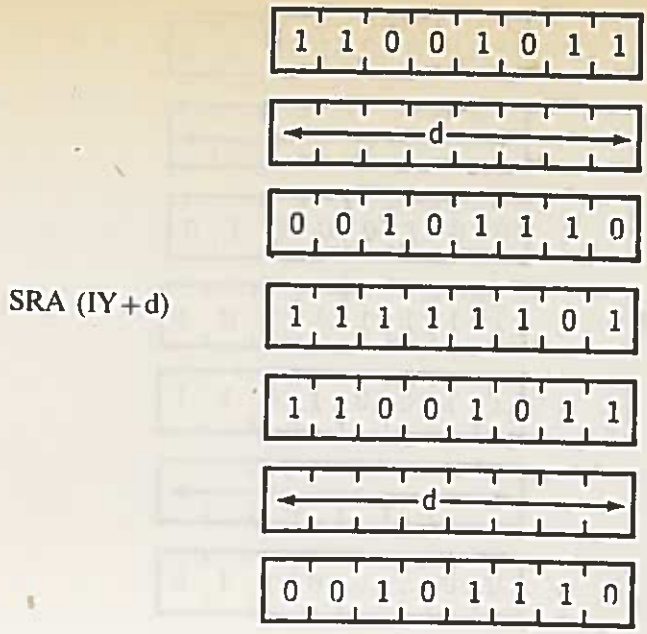
S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso
H: poner a «0»
N: poner a «0»
C: datos desde bit 7
P/V: poner a «1» si la paridad es par; poner a «0» en cualquier otro caso

SRA m



Un desplazamiento aritmético a la derecha se realiza en el contenido del operando m. El contenido del bit 0 se copia en el indicador de acarreo y el contenido anterior del bit 7 se mantiene invariable.



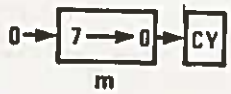


Instrucción	*Ciclos*	*Estados*
SRA r	2	8
SRA (HL)	4	15
SRA (IX+d)	6	23
SRA (IY+d)	6	23

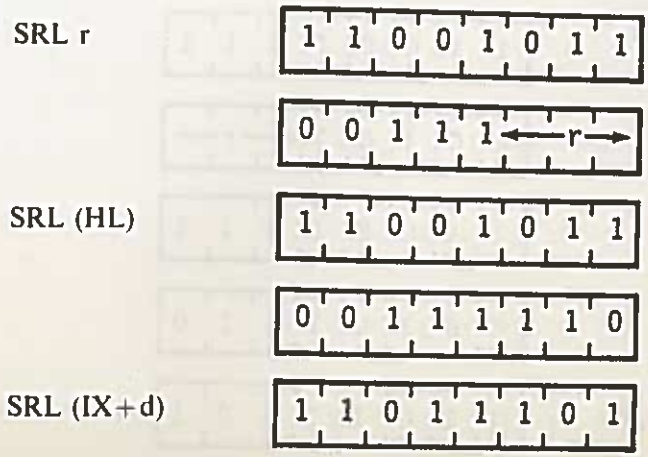
Indicadores: S, Z, H, N, C, P/V

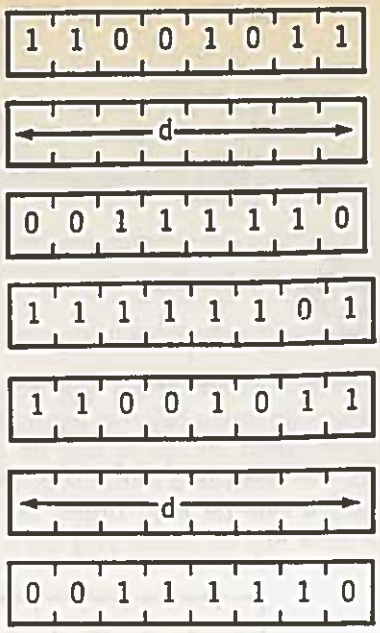
S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso
H: poner a «0»
N: poner a «0»
C: datos desde bit 0
P/V: poner a «1» si la paridad es par; poner a «0» en cualquier otro caso

SRL m



El contenido del operando m se desplaza a la derecha. El contenido del bit 0 se copia en el indicador de acarreo y se pone a «1» el bit 7.





SRL (IY+d)

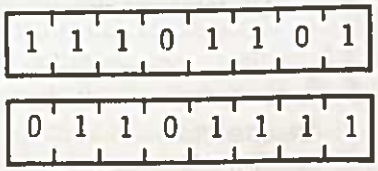
Instrucción	*Ciclos*	*Estados*
SRL r	2	8
SRL (HL)	4	15
SRL (IX+d)	6	23
SRL (IY+d)	6	23

Indicadores: S, Z, H, N, C, P/V

S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso
H: poner a «0»
N: poner a «0»
C: datos desde bit 0
P/V: poner a «1» si la paridad es par; poner a «0» en cualquier otro caso

RLD

El contenido de los 4 bits de orden bajo de posición de memoria (HL) se copian en los 4 bits de orden alto de esa misma posición de memoria. El contenido anterior de esos 4 bits de orden alto se copian en los 4 bits de orden bajo del acumulador y el contenido anterior de los 4 bits de orden bajo del acumulador se copian en los 4 bits de orden bajo de la posición de memoria (HL). No resulta afectado el contenido de los 4 bits de alto orden del acumulador.



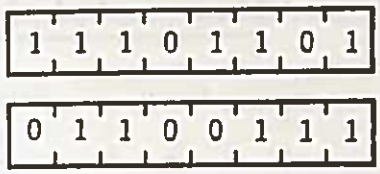
Ciclos: 5
Estados: 18
Indicadores: S, Z, H, N, P/V

S: poner a «1» si el acumulador es negativo después de la operación; poner a «0» en cualquier otro caso

Z: poner a «1» si el acumulador es 0 después de la operación; poner a «0» en cualquier otro caso
H: poner a «0»
N: poner a «0»
P/V: poner a «1» si la paridad del acumulador es par después de la operación; poner a «0» en cualquier otro caso

RRD

El contenido de los 4 bits de orden bajo de la posición de memoria (HL) se copia en los 4 bits de orden bajo del acumulador. El contenido anterior de estos últimos se copia en los 4 bits de alto orden de la posición de memoria (HL) y el contenido anterior de los 4 bits de alto orden de (HL) se copia en los 4 bits de bajo orden de (HL). No resulta afectado el contenido de los 4 bits de orden alto del acumulador.



Ciclos: 5
Estados: 18
Indicadores: S, Z, H, N, P/V

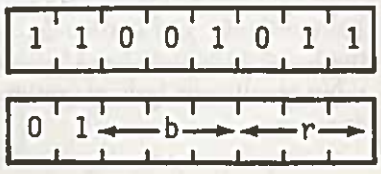
S: poner a «1» si el acumulador es negativo después de la operación; poner a «0» en cualquier otro caso
Z: poner a «1» si el acumulador es 0 después de la operación; poner a «0» en cualquier otro caso
H: poner a «0»
N: poner a «0»
P/V: poner a «1» si la paridad del acumulador es par después de la operación; poner a «0» en cualquier otro caso

GRUPO DE PRUEBA, PONER A «0» Y PONER A «1» BITS

BIT b, r

$Z \leftarrow \overline{r_b}$

Después de la ejecución de esta instrucción, el indicador Z en el registro F contendrá el complemento del bit indicado dentro del registro indicado.



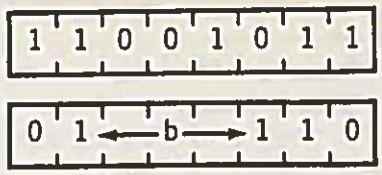
Ciclos: 2
Estados: 8
Indicadores: S, Z, H, N, P/V

S: desconocido
Z: poner a «1» si el bit especificado es 0; poner a «0» en cualquier otro caso
H: poner a «1»
N: poner a «0»
P/V: desconocido

BIT b, (HL)

$Z \leftarrow \overline{(HL)_b}$

Después de la ejecución de esta instrucción, el indicador Z en el registro F contendrá el complemento del bit indicado dentro del contenido del par de registros HL.



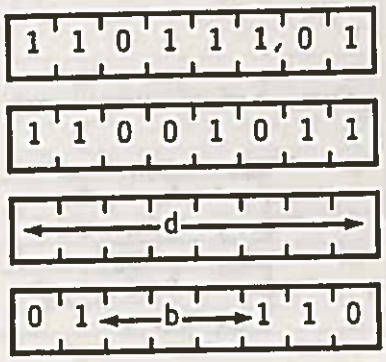
Ciclos: 3
Estados: 12
Indicadores: S, Z, H, N, P/V

S: desconocido
Z: poner a «1» si el bit especificado es 0; poner a «0» en cualquier otro caso
H: poner a «1»
N: poner a «0»
P/V: desconocido

BIT b, (IX+d)

$Z \leftarrow \overline{(IX+d)_b}$

Después de la ejecución de esta instrucción, el indicador Z en el registro F contendrá el complemento del bit indicado dentro del contenido de la posición de memoria apuntada por la suma del contenido del par de registros IX y el entero de desplazamiento d, de complemento a 2.



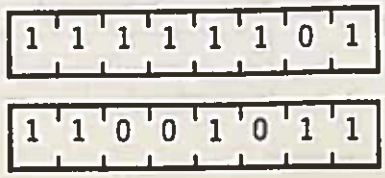
Ciclos: 5
Estados: 20
Indicadores: S, Z, H, N, P/V

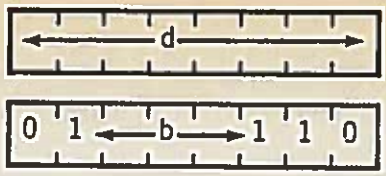
S: desconocido
Z: poner a «1» si el bit especificado es 0; poner a «0» en cualquier otro caso
H: poner a «1»
N: poner a «0»
P/V: desconocido

BIT b, (IY+d)

$Z \leftarrow \overline{(IY+d)_b}$

Después de la ejecución de esta instrucción, el indicador Z en el registro F contendrá el complemento del bit indicado dentro del contenido de la posición de memoria apuntada por la suma del contenido del par de registros IY y el entero de desplazamiento d, de complemento a 2.





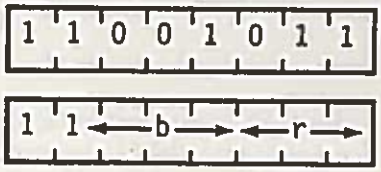
Ciclos: 5
Estados: 20
Indicadores: S, Z, H, N, P/V

S: desconocido
Z: poner a «1» si el bit especificado es 0; poner a «0» en cualquier otro caso
H: poner a «1»
N: poner a «0»
P/V: desconocido

SET b, r

$r_b \leftarrow 1$

Se pone a «1» el bit b (cualquier bit, de 7 a 0) en el registro r.

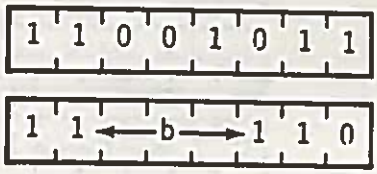


Ciclos: 2
Estados: 8
Indicadores: ninguno

SET b (HL)

$(HL)_b \leftarrow 1$

El bit b, en la posición de memoria direccionada por el contenido del par de registros HL, se pone a «1».

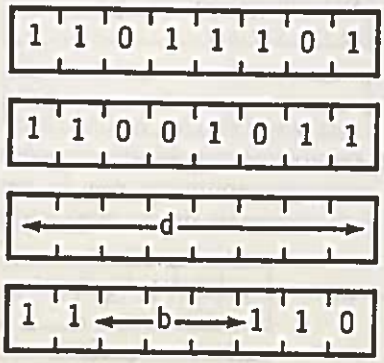


Ciclos: 4
Estados: 15
Indicadores: ninguno

SET b, (IX+d)

$(IX+d)_b \leftarrow 1$

Se pone a «1» el bit b en la posición de memoria direccionada por la suma del contenido del par de registros IX y el entero de desplazamiento de complemento a 2.

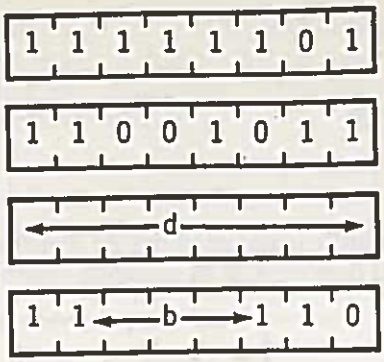


Ciclos: 6
Estados: 23
Indicadores: ninguno

SET b, (IY+d)

$(IY+d)_b \leftarrow 1$

Se pone a «1» el bit b en la posición de memoria direccionada por la suma del contenido del par de registros IY y el entero de desplazamiento de complemento a 2.



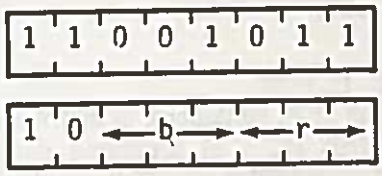
Ciclos: 6
Estados: 23
Indicadores: ninguno

RES b, m

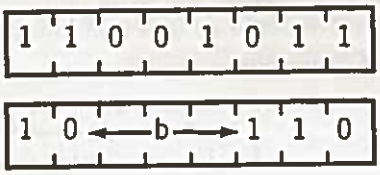
$s_b \leftarrow 0$

Se pone a «0» el bit b en el operando m.

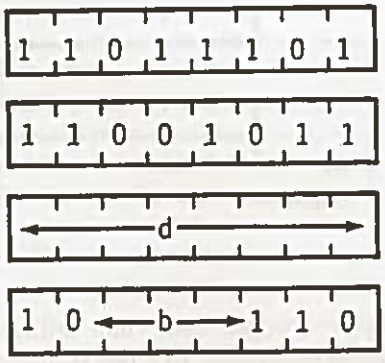
RES b, r



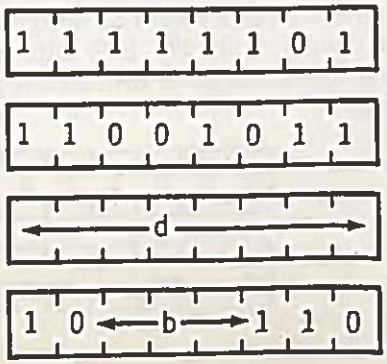
RES b, (HL)



RES b, (IX+d)



RES b, (IY+d)



Instrucción	*Ciclos*	*Estados*
RES b, r	4	8
RES b, (HL)	4	15
RES b, (IX+d)	6	23
RES b, (IY+d)	6	23

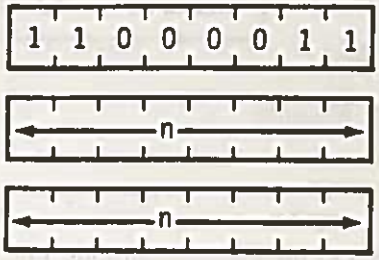
Indicadores: ninguno

GRUPO DE SALTOS

JP nn

PC←nn

El operando nn se carga en el par de registros PC (contador de programa) y apunta a la dirección de la siguiente instrucción de programa a ejecutar.

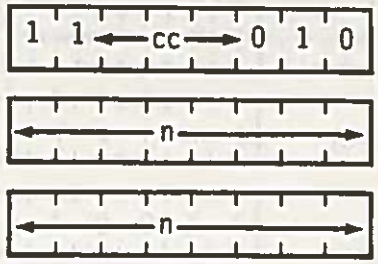


Ciclos: 3
Estados: 10
Indicadores: ninguno

JP cc, nn

IF cc TRUE, PC←nn

Si la condición cc es verdadera, la instrucción carga el operando nn en el par de registros PC y el programa continúa con la instrucción que comienza en la dirección nn. Si la condición cc es falsa, el contador de programa se incrementa en la forma habitual y el programa continúa con la siguiente instrucción secuencial.

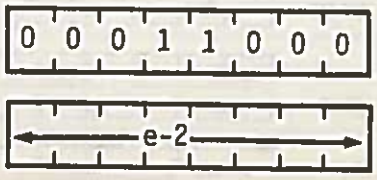


Ciclos: 3
Estados: 10
Indicadores: ninguno

JR e

PC←PC+e

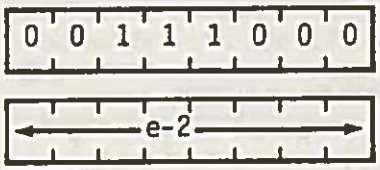
Esta instrucción proporciona una bifurcación incondicional a otros segmentos de un programa. El valor del desplazamiento e se añade al PC y la siguiente instrucción es objeto de búsqueda desde la posición designada por el nuevo contenido del PC. Este salto se mide a partir de la dirección del código de operación de la instrucción y tiene un margen de −126 a +129 octetos.



Ciclos: 3
Estados: 12
Indicadores: ninguno

JR C, e

Si C=0, continuar
Si C=1, PC←PC+e
Esta instrucción proporciona una bifurcación condicional a otros segmentos de programa que depende de los resultados de una prueba en el indicador de acarreo. Si el indicador se pone a «1», el valor del desplazamiento e se añade al PC y la siguiente instrucción es objeto de búsqueda a partir de la posición designada por el nuevo contenido del PC. Si el indicador se pone a «0», la siguiente instrucción se toma a partir de la posición que sigue a esta instrucción.



Si la condición se cumple:

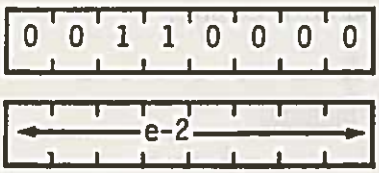
Ciclos: 3
Estados: 12

Si la condición no se cumple:

Ciclos: 2
Estados: 7
Indicadores: ninguno

JR NC, e

Si C=1, continuar
Si C=0, PC←PC+e
Esta instrucción proporciona una bifurcación condicional a otros segmentos de programa que depende de los resultados de una prueba en el indicador de acarreo. Si el indicador se pone a «0», el valor del desplazamiento e se añade al PC y la siguiente instrucción es objeto de búsqueda a partir de la posición designada por el nuevo contenido de PC. Si el indicador se pone a «1», la siguiente instrucción a ejecutar se toma de la posición que sigue a esta instrucción.



Si la condición se cumple:

Ciclos: 3
Estados: 12

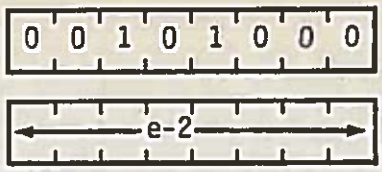
Si la condición no se cumple:

Ciclos: 2
Estados: 7
Indicadores: ninguno

JR Z, e

Si Z=0, continuar
Si Z=1, PC←PC+e
Si el indicador de cero se pone a «1», el valor del desplazamiento e se añade al PC y la siguiente instrucción es objeto de búsqueda a partir de la posición designada por el nuevo contenido del PC. Si se pone a «0» el

indicador de cero, la siguiente instrucción a ejecutar se extrae de la posición que sigue a esta instrucción.



Si la condición se cumple:

Ciclos: 3
Estados: 12

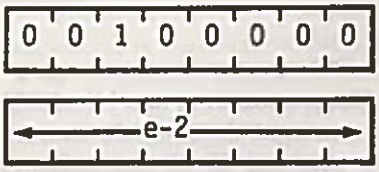
Si la condición no se cumple:

Ciclos: 2
Estados: 7
Indicadores: ninguno

JR NZ, e

Si Z=1, continuar
Si Z=0, PC←PC+e

Si el indicador de cero se pone a «0», el valor del desplazamiento e se añade al PC y la siguiente instrucción es objeto de búsqueda a partir de la posición designada por el nuevo contenido del PC. Si se pone a «1» el indicador de cero, la siguiente instrucción a ejecutar se extrae de la posición que sigue a esta instrucción.



Si la condición se cumple:

Ciclos: 3
Estados: 12

Si la condición no se cumple:

Ciclos: 2
Estados: 7
Indicadores: ninguno

JP (HL)

PC←HL

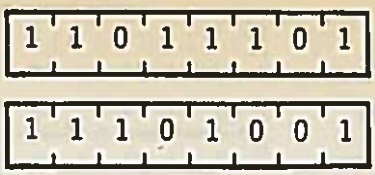
El PC se carga con el contenido del par de registros HL. La siguiente instrucción es objeto de búsqueda a partir de la posición designada por el nuevo contenido del PC.

Ciclos: 1
Estados: 4
Indicadores: ninguno

JP (IX)

PC←IX

El PC se carga con el contenido del par de registros IX. La siguiente instrucción es objeto de búsqueda a partir de la posición designada por el nuevo contenido del PC.



Ciclos: 2
Estados: 8
Indicadores: ninguno

JP (IY)

PC←IY

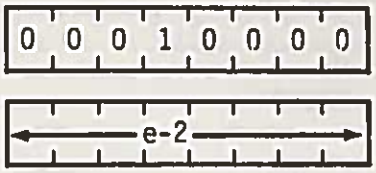
El PC se carga con el contenido del par de registros IY. La siguiente instrucción es objeto de búsqueda a partir de la posición designada por el nuevo contenido del PC.

Ciclos: 2
Estados: 8
Indicadores: ninguno

DJNZ, e

– – – – –

Se decrementa el registro B, y si persiste un valor distinto de 0, el valor del desplazamiento e se añade al PC. La siguiente instrucción es objeto de búsqueda a partir de la posición designada por el nuevo contenido del PC. Si como consecuencia del decremento se deja a B con un valor 0, la siguiente instrucción a ejecutar se toma de la posición que sigue a esta instrucción.



Si B≠0:

Ciclos: 3
Estados: 13

Si B=0:

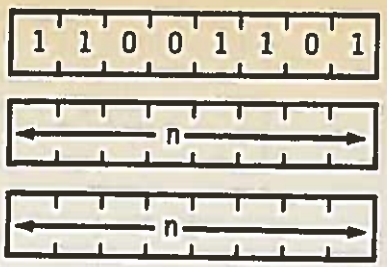
Ciclos: 2
Estados: 8
Indicadores: ninguno

GRUPO DE LLAMADA Y DE RETORNO

CALL nn

$(SP-1)\leftarrow PC_H$, $(SP-2)\leftarrow PC_L$, PC←nn

Después de llevar el contenido actual del PC a la parte superior de la pila de memoria externa, los operandos nn se cargan en PC para apuntar a la dirección en memoria en donde el primer código de operación de una subrutina ha de ser objeto de búsqueda. Observación: Puesto que se trata de una instrucción de 3 octetos, el PC se habrá incrementado en tres antes de que se ejecute dicha introducción en pila.

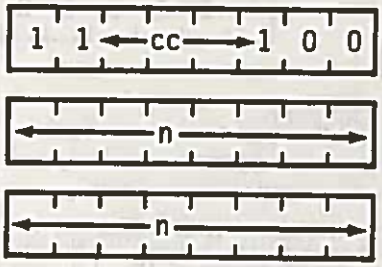


Ciclos: 5
Estados: 17
Indicadores: ninguno

CALL cc, nn

Si cc TRUE: $(SP-1)\leftarrow PC_H$, $(SP-2)\leftarrow PC_L$, $PC\leftarrow nn$

Si la condición cc es verdadera, esta instrucción lleva el contenido actual del PC a la parte superior de la pila de memoria externa y luego carga los operandos nn en PC para apuntar a la dirección en memoria en donde el primer código de operación de una subrutina ha de ser objeto de búsqueda.



Si cc es verdadera:

Ciclos: 5
Estados: 17

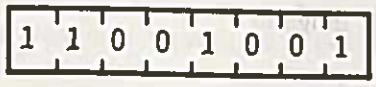
Si cc es falsa:

Ciclos: 3
Estados: 10
Indicadores: ninguno

RET

$PC_L\leftarrow(SP)$, $PC_H\leftarrow(SP+1)$

El control se vuelve al flujo de programa original extrayendo el contenido anterior del PC por la parte superior de la pila de memoria externa, adonde se llevó por la instrucción CALL. En el siguiente ciclo de máquina, el procesador central buscará el siguiente código de operación de programa a partir de la posición en memoria apuntada ahora por el PC.



Ciclos: 3
Estados: 10
Indicadores: ninguno

RET cc

Si cc VERDADERA: $PC_L\leftarrow(SP)$, $PC_H\leftarrow(SP+1)$

Si la condición cc es verdadera, el control se vuelve al flujo de programa original extrayendo el contenido anterior del PC de la parte superior de la pila de memoria externa, adonde se llevó por la instrucción CALL. En el siguiente ciclo de máquina, el procesador central buscará el siguiente código de operación de programa a partir de la posición en memoria ya apuntada por el PC. Si la condición cc es falsa, el PC se incrementa

simplemente en la forma habitual y el programa continúa con la siguiente instrucción secuencial.

Si cc es verdadera:

Ciclos: 3
Estados: 11

Si cc es falsa:

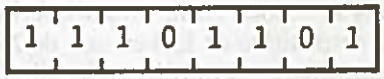
Ciclos: 1
Estados: 5
Indicadores: ninguno

RETI

Retorno desde interrupción
Esta instrucción se utiliza al final de una rutina de servicio de interrupción para:

1. Restaurar el contenido del PC.
2. Señalizar a un dispositivo de E/S que se ha completado la rutina de interrupción.

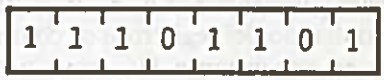
La instrucción RETI facilita la inclusión en cascada («anidado») de interrupciones que permite a los dispositivos de más alta prioridad suspender el servicio de rutinas de servicio de más baja prioridad. Esta instrucción pone a «0» también los flip-flops IFF1 y IFF2.



Ciclos: 4
Estados: 14
Indicadores: ninguno

RETN

Retorno desde interrupción no enmascarable
Utilizada al final de una rutina de servicio para una interrupción no enmascarable, la instrucción ejecuta un retorno incondicional que funciona de forma idéntica a la instrucción RET. El control se vuelve ahora al flujo de programa original; en el siguiente ciclo de máquina, el procesador central buscará el siguiente código de operación a partir de la posición en memoria ahora apuntada por el PC. Además, el estado de IFF2 se reproduce en IFF1 al estado que tuvo antes de la aceptación de NMI.

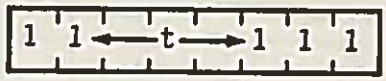


Ciclos: 4
Estados: 14
Indicadores: ninguno

RST p

$(SP-1) \leftarrow PC_H$ $(SP-2) \leftarrow PC_L$, $PC_H \leftarrow 0$, $PC_L \leftarrow p$
El contenido de PC actual se lleva a la pila de memoria externa y la posición de memoria de página cero dada por el operando p se carga en el

PC. A continuación, la ejecución de programa comienza con el código de operación en la dirección apuntada ahora por PC. La instrucción de reinicio permite un salto a una de 8 direcciones como se indica en la tabla siguiente. El operando p se ensambla en el código objeto con el empleo del estado t correspondiente.



p	t
00H	000
08H	001
10H	010
18H	011
20H	100
28H	101
30H	110
38H	111

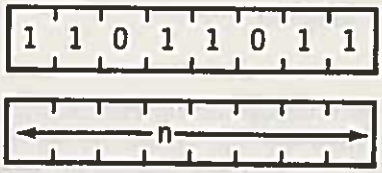
Ciclos: 3
Estados: 11
Indicadores: ninguno

GRUPO DE ENTRADA Y DE SALIDA

IN A, (n)

A←(n)

El operando n se coloca en la mitad inferior del bus de direcciones para seleccionar el dispositivo de E/S en uno de 256 ports posibles. El contenido del acumulador aparece también en la mitad superior del bus de direcciones en este instante. Un octeto procedente del port seleccionado se coloca entonces en el bus de datos y se escribe en el acumulador en el procesador central.

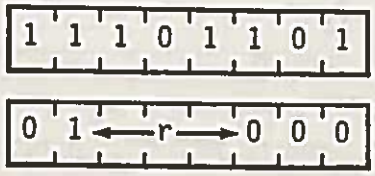


Ciclos: 3
Estados: 11
Indicadores: ninguno

IN r, (C)

r←(C)

El contenido del registro C se coloca en la mitad inferior del bus de direcciones para seleccionar el dispositivo de E/S en uno de los 256 ports posibles. El contenido del registro B se coloca en la mitad superior del bus de direcciones en este instante. Un octeto procedente del port seleccionado se coloca entonces en el bus de datos y se escribe en el registro r en el procesador central.



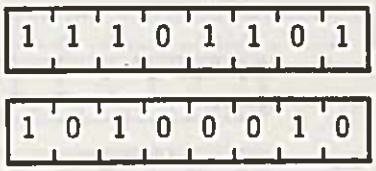
Ciclos: 3
Estados: 12
Indicadores: S, Z, H, N, P/V

S: poner a «1» si el resultado es negativo; poner a «0» en cualquier otro caso
Z: poner a «1» si el resultado es 0; poner a «0» en cualquier otro caso
H: poner a «0»
N: poner a «0»
P/V: poner a «1» si hay desbordamiento; poner a «0» en cualquier otro caso

INI

(HL)←(C), B←B−1, HL←HL+1

El contenido del registro C se coloca en la mitad inferior del bus de direcciones para seleccionar el dispositivo de E/S en uno de los 256 ports posibles. El registro B puede utilizarse como un contador de octetos y su contenido se coloca en la mitad superior del bus de direcciones. Un octeto procedente del port seleccionado se coloca entonces en el bus de datos y se escribe para el procesador central. El contenido del par de registros HL se coloca entonces en el bus de direcciones y el octeto de entrada es objeto de escritura en la posición de memoria correspondiente. Finalmente, se decrementa el contador de octetos y también se decrementa el par de registros HL.



Ciclos: 4
Estados: 16
Indicadores: S, Z, H, N, P/V
S: desconocido
Z: poner a «1» si B−1=0; poner a «0» en cualquier otro caso
H: desconocido
N: poner a «1»
P/V: desconocido

INIR

(HL)←(C), B←B−1, HL←HL+1

El contenido del registro C se coloca en la mitad inferior del bus de direcciones para seleccionar el dispositivo de E/S en uno de los 256 ports posibles. El registro B puede utilizarse como un contador de octetos y su contenido se coloca en la mitad superior del bus de direcciones. Un octeto procedente del port seleccionado se coloca entonces en el bus de datos y se escribe para el procesador central. El contenido del par de registros HL se coloca entonces en el bus de direcciones y el octeto de entrada es objeto de escritura en la posición de memoria correspondiente. Entonces, el contador de octetos se decrementa y el par de registros HL se incrementa. Si el decremento hace que B pase a 0, se terminará la instrucción. Si B no es 0, el PC se decrementa en 2 y se repite la instrucción. Las interrupciones se identificarán después de cada transferencia de datos.

Si B≠0:
Ciclos: 5
Estados: 21

Si B=0:

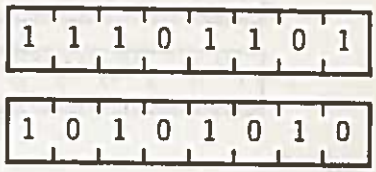
Ciclos: 4
Estados: 16
Indicadores: S, Z, H, N, P/V

S: desconocido
Z: poner a «1»
H: desconocido
N: poner a «1»
P/V: desconocido

IND

(HL)←(C), B←B−1, HL←HL−1

El contenido del registro C se coloca en la mitad inferior del bus de direcciones para seleccionar el dispositivo de E/S. El registro B puede utilizarse como un contador de octetos y su contenido se coloca en la mitad superior del bus de direcciones. Un octeto procedente del port seleccionado se coloca entonces en el bus de datos y se escribe para el procesador central. El contenido del par de registros HL se coloca entonces en el bus de direcciones y el octeto de entrada es objeto de escritura en la posición de memoria correspondiente. Finalmente, se decrementa el contador de octetos y también se decrementa el par de registros HL.



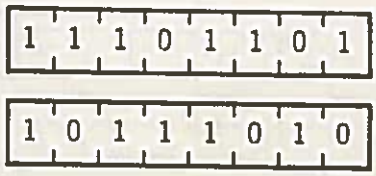
Ciclos: 4
Estados: 16
Indicadores: S, Z, H, N, P/V

S: desconocido
Z: poner a «1» si B−1=0; poner a «0» en cualquier otro caso
H: desconocido
N: poner a «1»
P/V: desconocido

INDR

(HL)←(C), B←B−1, HL←HL−1

El contenido del registro C se coloca en la mitad inferior del bus de direcciones para seleccionar el dispositivo de E/S. El registro B se utiliza como un contador de octetos y su contenido se coloca en la mitad superior del bus de direcciones. Un octeto procedente del port seleccionado se coloca en el bus de datos y se escribe para el procesador central. El contenido del par de registros HL se coloca en el bus de direcciones y el octeto de entrada se escribe en la posición de memoria correspondiente. El par de registros HL y luego el contador de octetos son objeto de decremento. Si el decremento hace que B pase a «0», el PC se decrementa en 2 y se repite la instrucción. Las interrupciones se identificarán después de cada transferencia de datos.



Si B≠0:

Ciclos: 5
Estados: 21

Si B=0:

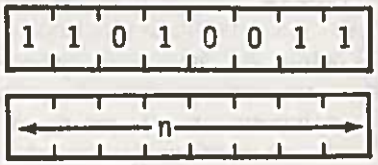
Ciclos: 4
Estados: 16
Indicadores: S, Z, H, N, P/V

S: desconocido
Z: poner a «1»
H: desconocido
N: poner a «1»
P/V: desconocido

OUT (n), A

(n)←A

El operando n se coloca en la mitad inferior del bus de direcciones para seleccionar el dispositivo de E/S. El contenido del acumulador aparece en la mitad superior del bus de direcciones. A continuación, el octeto contenido en el acumulador se coloca en el bus de datos y se escribe en el dispositivo periférico seleccionado.

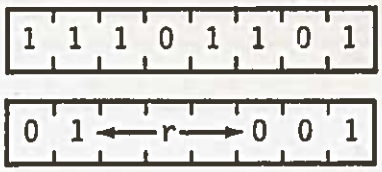


Ciclos: 3
Estados: 11
Indicadores: ninguno

OUT (C), r

(C)←r

El contenido del registro C se coloca en la mitad inferior del bus de direcciones para seleccionar el dispositivo de E/S. El contenido del registro B se coloca en la mitad superior del bus de direcciones. El octeto contenido en el registro r se coloca en el bus de datos y es objeto de escritura en el dispositivo periférico seleccionado.

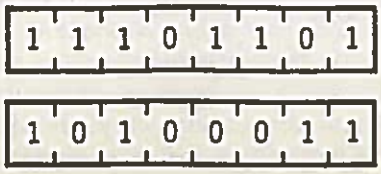


Ciclos: 3
Estados: 12
Indicadores: ninguno

OUTI

(C)←(HL), B←B−1, HL←HL+1

El contenido del par de registros HL se coloca en el bus de direcciones para seleccionar una posición en memoria. El octeto contenido en esta posición de memoria se almacena temporalmente en el procesador central. Una vez decrementado el contador de octetos (B), el contenido del registro C se coloca en la mitad inferior del bus de direcciones para seleccionar el dispositivo de E/S. El registro B puede emplearse como un contador de octetos y su valor decrementado se coloca en la mitad superior del bus de direcciones. El octeto que ha de ser objeto de salida se coloca en el bus de datos y es objeto de escritura en el dispositivo periférico seleccionado. Finalmente, se incrementa el par de registros HL.



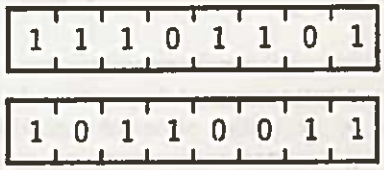
Ciclos: 4
Estados: 16
Indicadores: S, Z, H, N, P/V

S: desconocido
Z: poner a «1» si B−1=0; poner a «0» en cualquier otro caso
H: desconocido
N: poner a «1»
P/V: desconocido

OTIR

(C)←(HL), B←B−1, HL←HL+1

El contenido del par de registros HL se coloca en el bus de direcciones para seleccionar una posición de memoria. El octeto contenido en esta posición de memoria se almacena temporalmente en el procesador central. Una vez decrementado el contador de octetos (B), el contenido de registro C se coloca en la mitad inferior del bus de direcciones para seleccionar el dispositivo de E/S. El registro B puede emplearse como un contador de octetos y su valor decrementado se coloca en la mitad superior del bus de direcciones. El octeto que ha de ser objeto de salida se coloca en el bus de datos y es objeto de escritura en el dispositivo periférico seleccionado. A continuación, se incrementa el par de registros HL. Si el registro B decrementado no es 0, el PC se decrementa en 2 y se repite la instrucción. Si B es 0, se termina la instrucción. Las interrupciones se identificarán después de cada transferencia de datos.



Si B≠0:

Ciclos: 5
Estados: 21

Si B=0:

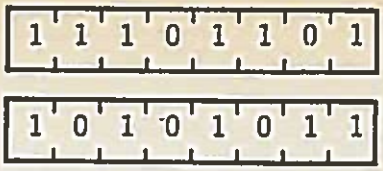
Ciclos: 4
Estados: 16
Indicadores: S, Z, H, N, P/V

S: desconocido
Z: poner a «1»
H: desconocido
N: poner a «1»
P/V: desconocido

OUTD

(C)←(HL), B←B−1, HL←HL−1

El contenido del par de registros HL se coloca en el bus de direcciones para seleccionar una posición de memoria. El octeto contenido en esta posición de memoria se almacena temporalmente en el procesador central. Una vez decrementado el contador de octetos (B), el contenido del registro C se coloca en la mitad inferior del bus de direcciones para seleccionar el dispositivo de E/S. El registro B puede emplearse como un contador de octetos y su valor decrementado se coloca en la mitad superior del bus de direcciones. El octeto que ha de ser objeto de salida se coloca en el bus de datos y es objeto de escritura en el dispositivo periférico seleccionado. Finalmente, se incrementa el par de registros HL.



Ciclos: 4
Estados: 16
Indicadores: S, Z, H, N, P/V
S: desconocido
Z: poner a «1» si B−1=0; poner a «0» en cualquier otro caso
H: desconocido
N: poner a «1»
P/V: desconocido

OTDR

(C)←(HL), B←B−1, HL←HL−1

El contenido del par de registros HL se coloca en el bus de direcciones para seleccionar una posición en memoria. El octeto contenido en esta posición de memoria se almacena temporalmente en el procesador central. Una vez decrementado el contador de octetos (B), el contenido del registro C se coloca en la mitad inferior del bus de direcciones para seleccionar el dispositivo de E/S. El registro B puede emplearse como un contador de octetos y su valor decrementado se coloca en la mitad superior del bus de direcciones. El octeto que ha de ser objeto de salida se coloca en el bus de datos y es objeto de escritura en el dispositivo periférico seleccionado. A continuación, se decrementa el par de registros HL. Si el registro B decrementado no es 0, el PC se decrementa en 2 y se repite la instrucción. Si el registro B es 0 entonces se termina la instrucción. Las interrupciones se identificarán después de cada transferencia de datos.

Si B≠0:

Ciclos: 5
Estados: 21

Si B=0:

Ciclos: 4
Estados: 16
Indicadores: S, Z, H, N, P/V
S: desconocido
Z: poner a «1»
H: desconocido
N: poner a «1»
P/V: desconocido

Capítulo 4

CONSTRUYA SU PROPIA COMPUTADORA. Comience con los elementos básicos

La computadora a construir a partir del diseño descrito en este libro se denomina ZAP, como abreviatura de Z80 Applications Processor. Construir una computadora desde el principio es, al mismo tiempo, educativo y utilitario (y se ahorra dinero). Explico detalladamente cada sección del proceso de construcción. Cada paso debe ser probado, preferiblemente antes de proseguir con la siguiente etapa. Aun cuando ello no es posible en todos los casos, hay un beneficio al seguir este camino. A menudo los buenos diseños no funcionan adecuadamente porque el nivel de la construcción está más allá de la capacidad del constructor.

Se supone que la mayoría de los aficionados de la electrónica no poseen equipos de prueba sofisticados, tales como osciloscopios o analizadores lógicos y, por tanto, he mantenido los procedimientos de prueba con la forma más simple posible. Con la división de ZAP en secuencias lógicas para revisión y prueba (y empleando componentes comprobados), los problemas pueden ser identificados en etapas tempranas y ser resueltos más fácilmente.

La implementación inicial de ZAP constituirá una configuración operable mínima. Y es importante que funcione adecuadamente antes de que intente añadir algo de los periféricos opcionales. Se hará todo lo posible para familiarizar al lector con los componentes y con la filosofía del diseño. Aunque sea necesario montar todos los componentes de esta configuración básica con el fin de revisar el funcionamiento correcto del procesador central, la comprobación previa del submontaje completo debe corregir (así lo espero) cualquier error de conexión.

La parte básica del ZAP está dividida en cuatro submontajes principales: La configuración de buses y control del Z80, la memoria y el decodificador de selección del chip E/S, la memoria y los registros de entrada/salida. Estas divisiones principales están subdivididas a nivel de los componentes. Los diagramas incluyen una explicación completa de su función lógica, y los procedimientos de prueba están descritos después de cada presentación de construcción.

EL PROCESADOR

La figura 4.1 es un diagrama de bloques detallado de la parte básica de la computadora ZAP.

I. Configuración de buses del Z80 y su lógica de control

A. Generación del reloj

La computadora ZAP funciona sobre la base de un reloj de 2,5 MHz TTL. A diferencia con el 8080A, el Z80 solamente requiere un reloj de fase simple y puede ser excitado con c.c a 2,5 MHz (el Z80A funciona a 4 MHz). En la figura 4.2 se ilustra el ciclo básico de temporización de la computadora.

Cada operación básica (M_N) de la computadora es completada en tres o seis períodos del reloj. La figura 4.2 muestra un ciclo de instrucción típico que consiste en tres ciclos de máquina: Búsqueda, lectura de memoria y escritura de memoria. Después de que se busque el código de operación (opcode) durante M1, los ciclos siguientes desplazan los datos entre la memoria y el procesador central.

Las figuras 4.3a y 4.3b ilustran dos diseños posibles para el Z80. Ambos circuitos de reloj tienen una resistencia de activación de 330 ohmios para +5 V. Ello satisface las necesidades de señales de reloj de c.a y c.c, pero es mejor emplear una sección de puerta inversora independiente para excitar la resistencia de activación sea cual fuere la técnica de oscilación.

El circuito controlado por cristal de la figura 4.3a es preferible si ha de mantenerse un tiempo de ejecución constante. De este modo, el circuito de la figura 4.3b, aunque sea aceptable, debe evitarse si la computadora ha de utilizarse como temporizador de sucesos. Puede servir para un propósito muy útil en las etapas de desarrollo, sin embargo, permitiendo al usuario hacer más lento el reloj (aumentando los valores de R y de C) hasta un ritmo en el que sea posible controlar directamente el funcionamiento del procesador central. Si todavía fuera necesario establecer un paso único del reloj, debe utilizarse el circuito de la figura 4.4. Dados los múltiples ciclos de reloj necesarios para ejecutar una sola instrucción, se precisarían muchos pulsadores para seguir un programa a través de su ejecución.

Un método de diagnóstico mucho más fácil sería emplear un circuito de paso único de instrucción. El circuito mostrado en la figura 4.5 no es parte del esquema final del ZAP, porque es necesario solamente si el constructor tiene un problema y necesita seguir la ejecución de un programa instrucción por instrucción. Esta función de paso único se realiza utilizando las señales de control generadas por el Z80 durante la ejecución del programa. Las dos señales particulares de interés son $\overline{M1}$ y $\overline{WAIT}$. $\overline{M1}$ es una salida y $\overline{WAIT}$ es una entrada. Como se muestra en la figura 4.6, $\overline{M1}$ va a un nivel lógico «0» al comienzo de cada ciclo de búsqueda de instrucción. $\overline{M1}$ significa que la computadora ha completado una instrucción y está comenzando con la siguiente. El objetivo es detener el microprocesador antes de que ejecute esta siguiente instrucción.

La entrada $\overline{WAIT}$ del Z80 hace justamente eso. Un nivel lógico «0» aplicado a esta entrada suspenderá la ejecución del programa de la computadora y lo mantendrá indefinidamente en el ciclo $\overline{M1}$. Durante T_2, el procesador central muestra la línea de entrada $\overline{WAIT}$ con el flanco posterior del reloj. Si, en este tiempo, $\overline{WAIT}$ está a un nivel lógico «0», un estado adicional de espera será introducido y la línea será muestreada de nuevo. El procesador central se mantendrá en esta modalidad hasta que $\overline{WAIT}$ pase a un nivel lógico «1». Se debe observar que no se trata de una orden de parada de la computadora.

El propósito real detrás de estas señales es permitir que se utilice la memoria relativamente lenta y periféricos con un procesador central muy rápido. Estados de espera adicionales deben ser introducidos sólo cuando sea necesario que el procesador central acceda a estos dispositivos. El efecto es sincronizar la temporización entre el procesador central y sus dispositivos de E/S. El circuito de la figura 4.5 nos permite controlar el estado $\overline{WAIT}$ y ejecutar solamente una instrucción con cada pulsación del botón. La salida en CI 1, terminal 8 (la entrada $\overline{WAIT}$), es normalmente baja, dando lugar a una espera indefinida. Cuando el botón es pulsado, un pulso único con supresión de rebotes sincroniza a CI 2, que es un circuito (flip-flop) tipo D. La duración de este pulso (el tiempo que mantiene el botón oprimido) carece de importancia, porque el circuito flip-flop es disparado por el flanco y solamente está relacionado con el flanco de entrada. Al pulsar el botón se activa CI 2 y se eleva el nivel de la línea $\overline{WAIT}$. Ya no se le dice que espere y el procesador central ejecuta la instrucción a plena velocidad del reloj. Como está a punto de empezar el siguiente ciclo de búsqueda de instrucción, $\overline{M1}$ pasa a nivel bajo como antes y dispara el circuito conectable.

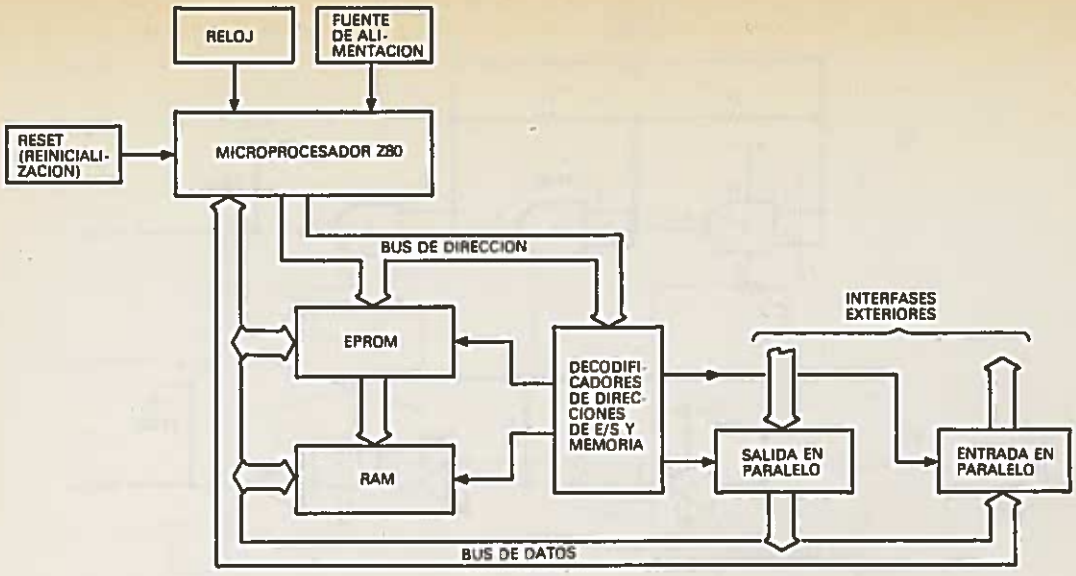


Figura 4.1 *Un diagrama de bloques de un sistema ZAP mínimo.*

Cuando se activa CI 3 reinicializa a CI 2 y vuelve el procesador central a una condición de espera hasta que sea pulsado el botón la siguiente vez.

La característica de paso único no es de mucho uso en una computadora a menos que haya alguna forma de controlar el contenido de todos los registros y de determinar lo que está tratando de hacer la computadora en cualquier momento. Para efectuar esto, el ZAP debe ser completamente operacional y estar ejecutando un programa monitor de punto de ruptura («breakpoint») que permita al usuario efectuar un paso simple con una rutina de software. Examinaremos estos programas más adelante.

Este hecho es de poca consolación para una persona con una computadora con depuración parcial de errores, o error de hardware que mantiene en programas largos de seguimiento lateral. Aunque convendría ver todos los contenidos de los registros, es prácticamente imposible sin tener un procesador central que pueda ejecutar una rutina de vaciado y de visualización. Ello no puede hacerse empleando el circuito de escalonamiento de hardware de la figura 4.5. Es posible, no obstante, examinar los contenidos de los buses de direcciones y de datos mientras el procesador central está parado. Esto debe dar una buena indicación de si la computadora está trabajando adecuadamente.

Muchos instrumentos pueden utilizarse para leer los niveles TTL en los buses. Un osciloscopio o un voltímetro de alta impedancia puede ser empleado, pero una presentación visible del contenido del bus es una mejor idea. Los circuitos en la figura 4.7 muestran métodos simples para presentar los contenidos de los buses de direcciones y de datos. Los circuitos están incluidos como auxiliares y no son necesarios para la operación del ZAP.

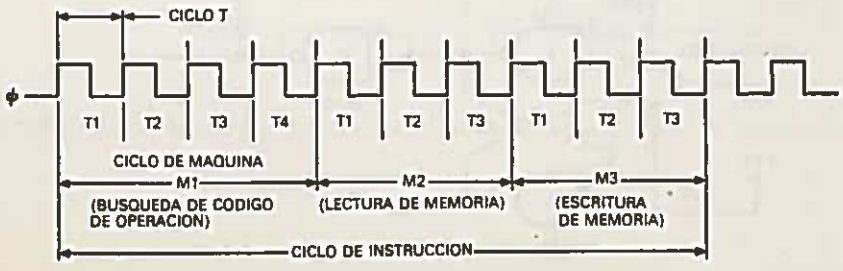


Figura 4.2 *Un ejemplo de temporización (cronograma) durante un ciclo típico de instrucción.*

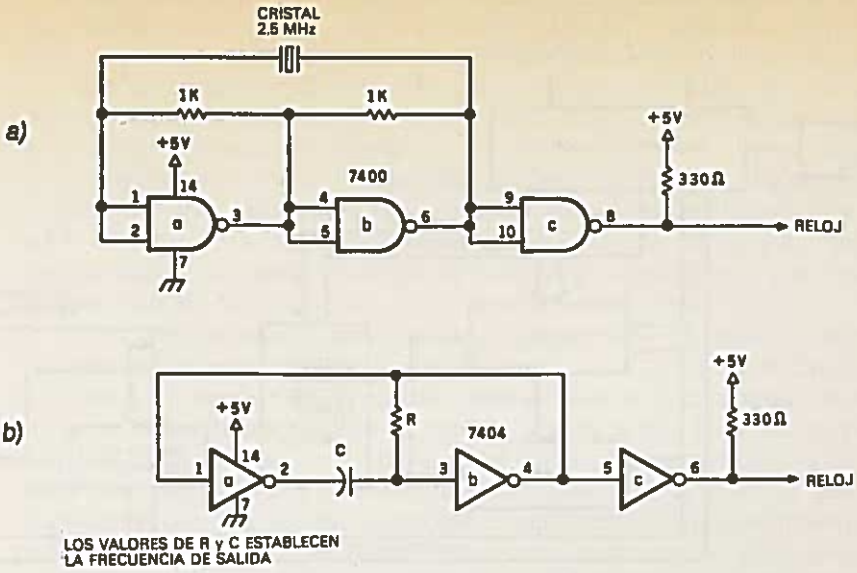


Figura 4.3 *Circuitos típicos de reloj de 2,5 MHz para el Z80.*
a) Control de cristal.
b) Con un oscilador de frecuencia variable.

Básicamente, el circuito de la figura 4.7a es un excitador de LED simple que está duplicado dieciséis veces para el bus de direcciones y ocho veces para el bus de datos. Puesto que el Z80 debe excitar solamente una carga TTL desde cada terminal de salida (las entradas de los excitadores de buses ya están unidas), algunos excitadores de display de este tipo deben estar unidos en el lado de la salida de los excitadores de buses. Este circuito servirá como un panel frontal elemental para cualquier constructor que sienta que una computadora no está completa sin luces intermitentes.

A veces surge la necesidad de controlar un punto simple en un circuito y de vigilar los cambios de nivel. Aunque el excitador de LEDs de la figura 4.7a detectará un cambio lento de nivel, prescindiría de pulsos cortos tales como $\overline{MI}$. Para controlar la ocurrencia de tales sucesos, especialmente si no hay disponible un osciloscopio para hacer las pruebas, es recomendable construir el circuito de la figura 4.7b. Esta sonda lógica simple es adecuada para muchas aplicaciones, pero se deben tomar precauciones en su empleo. No puede detectar un circuito abierto y el detector de pulsos sólo se dispara en el flanco negativo de cualquier transición. Si se plantearan algunos problemas, hay que añadir el circuito opcional con el empleo del 7486; ello permitirá detectar uno u otro flanco.

La sonda lógica o un detector de nivel lógico similar (osciloscopio, voltímetro digital, polímetro, etc.) es necesario para pruebas estáticas de los submontajes.

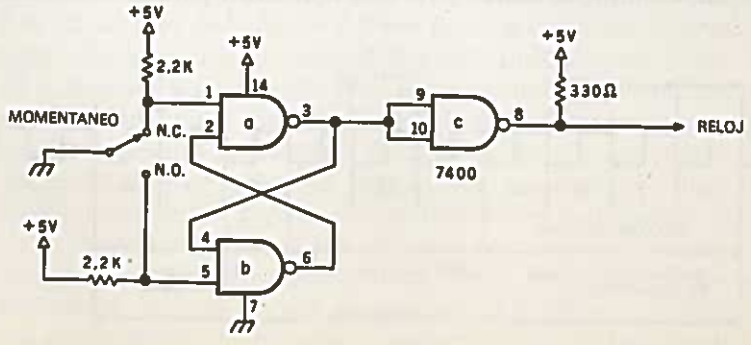


Figura 4.4 *Un circuito de generador de reloj para un ciclo simple.*

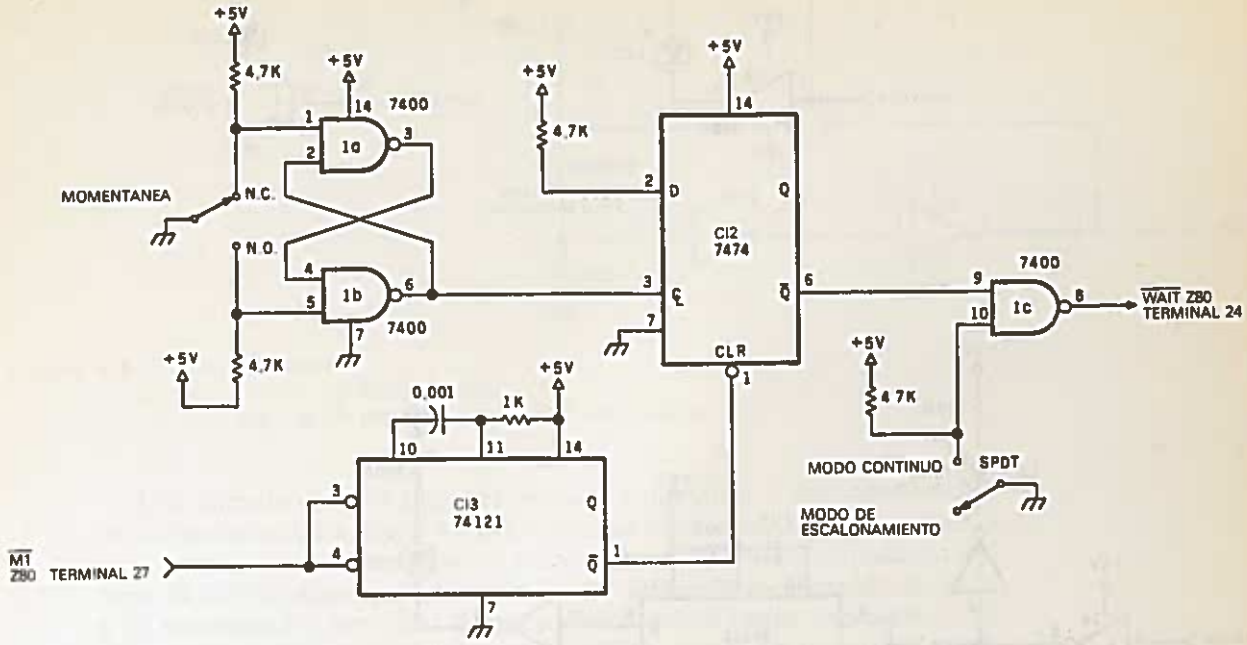


Figura 4.5 *Un circuito de paso simple de instrucción.*

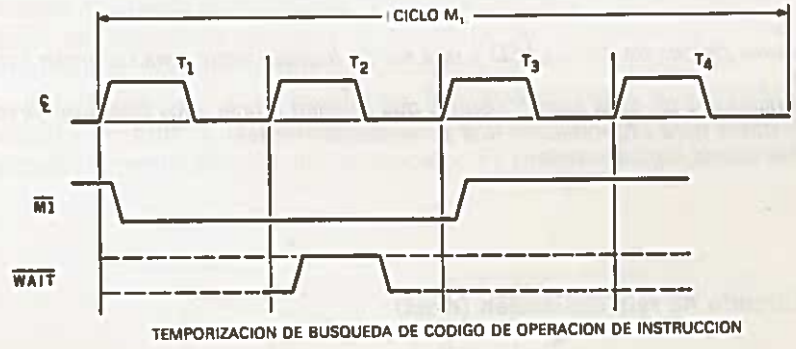


Figura 4.6 *Cronograma de búsqueda de código de operación ($\overline{M1}$) de instrucción.*

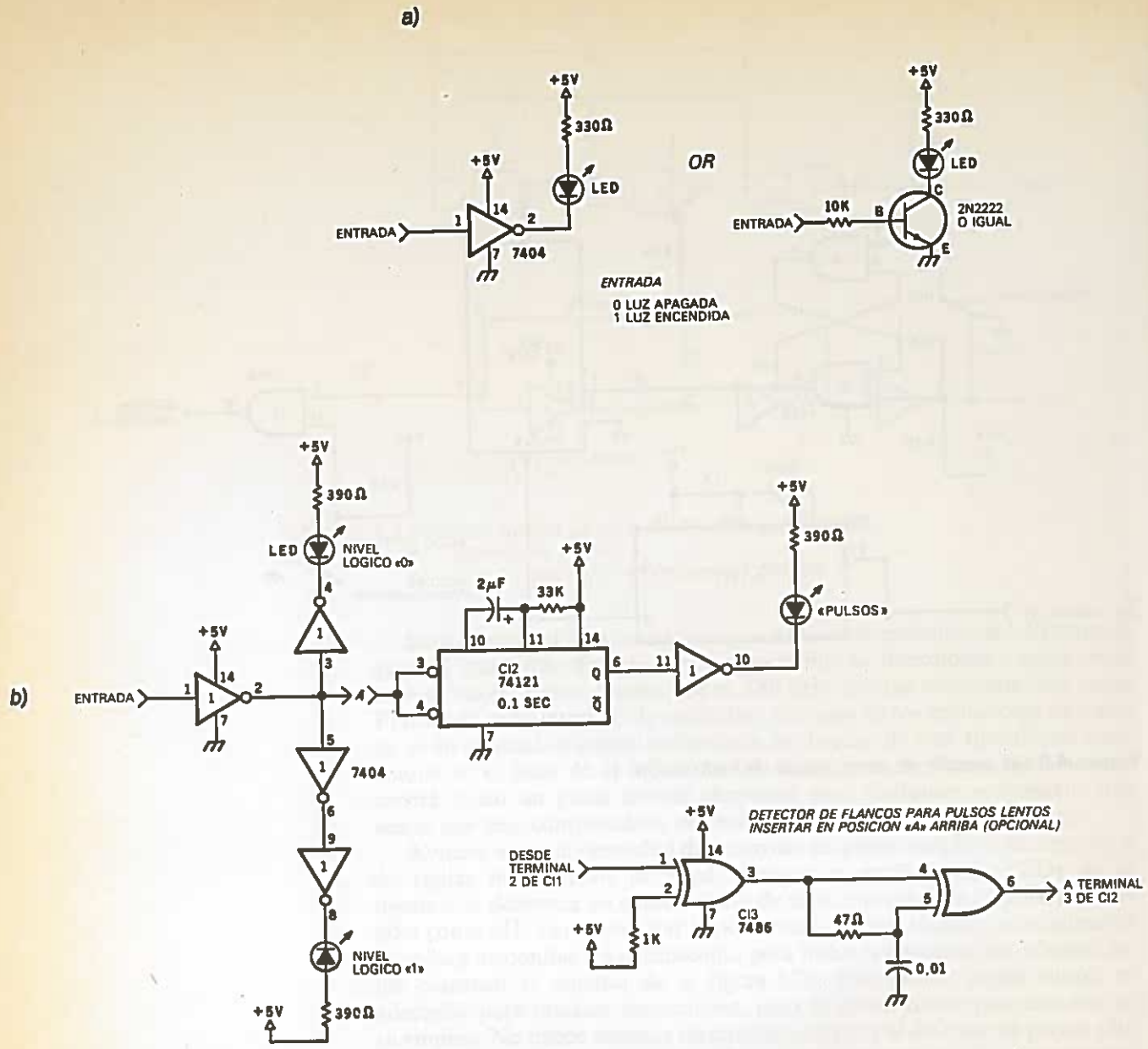


Figura 4.7 *Excitadores típicos de diodos LED y una sonda lógica simple para controlar los cambios de niveles lógicos.*

a) Indicadores de nivel lógico visibles que pueden unirse a los buses de direcciones y de datos para proporcionar una presentación visual.

b) Una sonda lógica simple.

B. Circuito de reinicialización (reset)

Casi siempre ignorada la función de reinicialización, es uno de los controles más necesarios de una computadora. Su importancia es reconocida inmediatamente cuando se ejecuta un programa en forma incorrecta. La orden de reinicialización (reset) en el Z80 interrumpe la ejecución y carga el contador de programa con 00 hexadecimal (la más baja dirección de memoria). Ello permite al programador reiniciar el programa. Cuando se combina con el circuito de paso único de instrucción anteriormente descrito, los programas pueden comenzarse, detenerse y volverse a iniciar en cualquier momento.

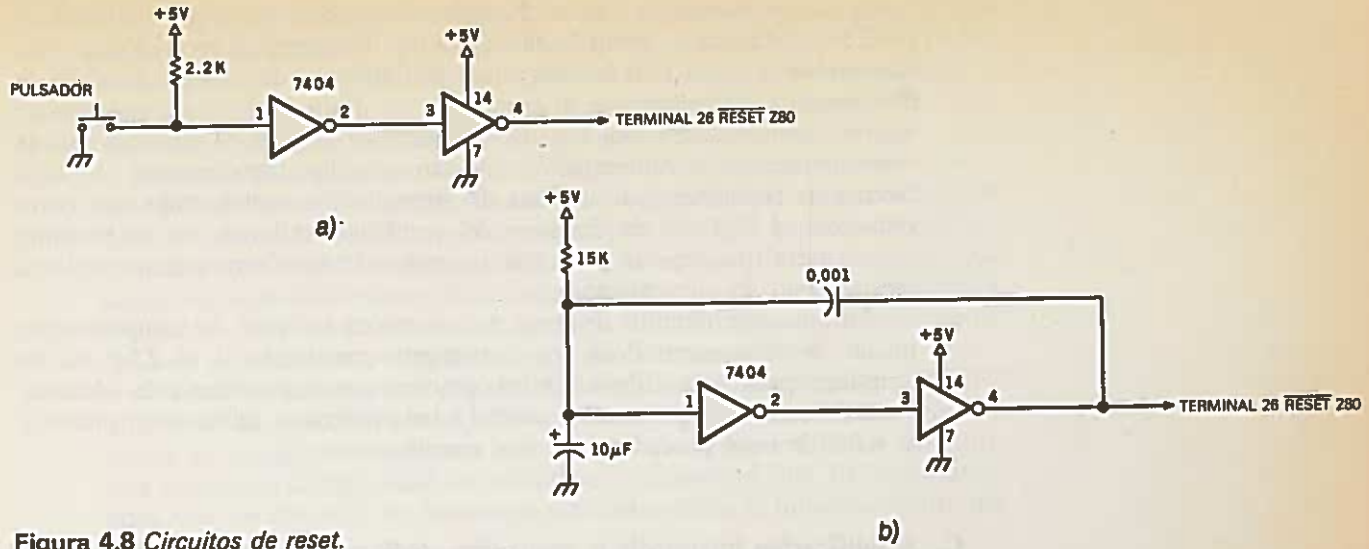


Figura 4.8 *Circuitos de reset.*
a) Un circuito de reset manual.
b) Un circuito de reset de encendido automático.

Una entrada de reset suele ser manual, automática o una combinación de ambas formas. La figura 4.8 es un circuito normal de «push to reset». Su salida suele ser alta hasta que se pulsa el botón y entonces pasa a nivel bajo. El Z80 permanecerá en reset mientras se mantenga oprimido el botón y no comenzará a funcionar de nuevo hasta que el botón sea liberado. El reset manual es una necesidad para la revisión inicial del programa, y este circuito es utilizado en el ZAP básico.

Cuando las computadoras se usan en aplicaciones en donde la presencia humana no interviene, como un controlador de luces de tráfico, el reset manual no puede emplearse y debe ser utilizado en cambio un reset automático. La figura 4.8b es el circuito de un reset de puesta en funcionamiento completamente automático. Cuando la alimentación se aplica primero a la computadora, el condensador de 10 mF se descargará completamente. El nivel lógico «0» resultante en la entrada del terminal 1 del 7404 se mantendrá durante unos 50 ms, mucho después de que se haya alimentado con +5 V el resto de la computadora. El régimen largo de carga del condensador generará, a su vez, a un nivel lógico «0» (una condición de reset) a la computadora hasta que el nivel de entrada se eleve a aproximadamente 2 V (un nivel lógico «1» TTL). Cuando se aplica la potencia total, el tiempo que tarda el circuito de reset en alcanzar 2 V constituirá un pulso de $\overline{\text{Reset}}$ de encendido de unos 35 ms. Para que la máquina volviera a sus condiciones iniciales se precisaría desconectar la alimentación.

En la figura 4.9 se combinan los reset manual y automático. Este circuito permite a la computadora comenzar la ejecución del programa inmediatamente después del encendido. El programa puede ser detenido y

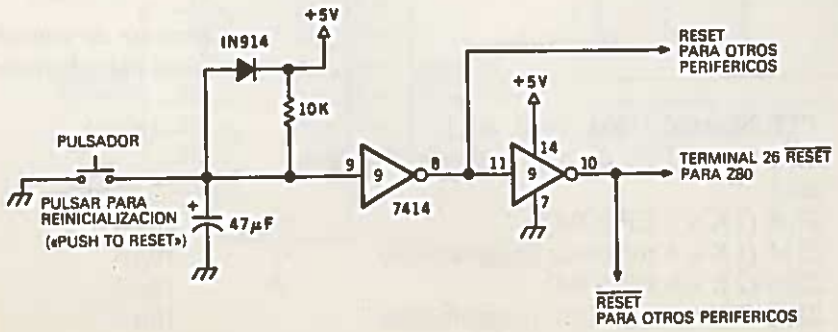


Figura 4.9 *Un circuito para combinar las funciones manuales y automáticas de reset.*

reiniciado pulsando el botón de reset. Componentes algo diferentes y funciones adicionales están incluidas en este diagrama. Los inversores con disparador Schmitt (7414s) aumentan la fiabilidad del diseño. Cuando se desconecta la alimentación, el empleo de un diodo para descargar rápidamente el condensador asegura que se genere un pulso si se vuelve a aplicar repentinamente la alimentación. Puesto que las interferencias de baja frecuencia («glitches») de la línea de alimentación suelen tener una corta duración, el régimen de descarga del condensador ha de ser lo bastante rápido para que deje de generarse un pulso de reset una vez que se haya reestablecido la alimentación.

Aunque este circuito de reset no sea necesario para la comprobación inicial de la computadora, es conveniente emplearle si el ZAP ha de ampliarse para incluir alguna de las opciones que se describen más adelante. Para sincronizar el procesador central y los periféricos, deben estar unidos a la señal de reset procedente de este circuito.

C. Amplificación intermedia o separación «buffering» de salidas de control y bus de direcciones

El Z80 tiene la capacidad necesaria para direccionar directamente 65.536 (que suele denominarse 64 K) octetos individuales de memoria de programa y 256 ports de entrada y de salida individuales. Puesto que el microprocesador es un dispositivo binario, es natural que esta dirección sea binaria. Hay 16 líneas de direcciones binarias con las etiquetas A0 a A15 inclusive. A0 es el LSB (bit menos significativo) y A15 es el MSB (bit más significativo).

Los niveles lógicos en este bus no son arbitrarios. La sección de control del procesador central activa el contador del programa para la próxima instrucción a ser ejecutada y en el ciclo de búsqueda coloca el contenido del contador del programa en el bus de direcciones. Durante las instrucciones de E/S, los ciclos de temporización adicionales colocan la dirección del dispositivo de E/S en los 8 bits menos significativos (A0 a A7 inclusives). Como este bus tiene que controlar las entradas de muchos dispositivos en paralelo, todos los cuales consumen alguna potencia de entrada, el bus de direcciones debe tener una corriente de salida que satisfará la demanda de carga. El Z80, en sí mismo, puede absorber un máximo de 1,8 mA o una carga TTL en cada terminal. Ello no constituye problema alguno si el diseñador utiliza memorias de baja potencia y chips de interfases de periféricos. Se trata de dispositivos caros y su uso no serviría necesariamente para instruir al constructor de la misma forma que las configuraciones de circuitos menos complejos.

El empleo de dispositivos TTL y CI de densidad más baja para funciones de decodificación es menos caro, pero exige considerablemente más alimentación procedente del bus. En la siguiente tabla indica la carga de entrada de diversos dispositivos:

Dispositivo	*Corriente de entrada en el caso más desfavorable*
TTL Normal (7404, 7442, etc.)	1,6 mA
Schottky y TTL de baja potencia (74LS04, etc.)	0,18 mA
2708 (1 K × 8 EPROM)	10 μA
2114 (1 K × 4 memoria programable)	10 μA
2716 (2 K × 8 EPROM)	10 μA
2102 (1 K × 1 memoria programable)	10 μA
8212 (enclavador [latch] de 8 bits)	0,25 mA
8T97 (controlador de 6 bits)	1,0 mA

Es fácil constatar que los consumidores de potencia real son dispositivos TTL. Los dispositivos Schottky TTL de baja potencia (LSTTL) pueden sustituir en todo el ZAP. Ahorran potencia a un coste algo mayor, pero el circuito tiene suficiente potencia para alimentar los dispositivos TTL. Si se sustituye LSTTL, se debe hacer tal sustitución en todo el sistema.

La carga producida por memoria, sobre todo con sólo 2 koctetos en la mitad ZAP básica, es insignificante. Con una corriente de excitación de 1,8 mA disponible del Z80, podríamos emplear LSTTL para la decodificación de direcciones de memoria y E/S, pero habría que limitar el abanico de salida «fanout» (conexiones de entrada totales) en cada línea de dirección a 9 entradas LSTTL. Ello es suficiente para el ZAP básico y probablemente sería un procedimiento aceptable, pero no es recomendable.

Cuando un usuario une, por primera vez, la sonda lógica (figura 4.7b) a una línea de dirección sin «buffer», la computadora puede sufrir graves daños. La carga presentada por la sonda, así como por los demás circuitos, será superior a la capacidad de excitación o control del bus. Es importante, pues, que los dispositivos de control no obstaculicen el funcionamiento del circuito.

Mejor que intentar optimizar el diseño hasta un grado que obligue al usuario a considerar cada μA (microamperio) consumido por sondas de prueba y excitadores de LEDs, es más fácil añadir «buffers» (amplificadores separadores o intermedios) que aumentar la potencia de salida del bus hasta un punto en el que la carga no sea un factor importante. Esta es la filosofía que hay detrás de la configuración de buses del ZAP y, como un beneficio secundario, proporcionará suficiente potencia para ampliar ZAP a 64 K si el usuario quisiera hacerlo así. También permite al usuario añadir sus propios circuitos TTL sin llegar a preocuparse excesivamente por la carga del bus.

Para conseguir una salida de alta potencia a partir del bus de direcciones, se utiliza un dispositivo de amplificación intermedia «buffering» (denominado controlador de bus no inversor). Las salidas A0 a A15 inclusives del Z80 sólo tienen una conexión, que es la de entrada de los controladores. Todos los demás dispositivos que utilizan la dirección están unidos a la salida de los controladores.

La figura 4.10 es el diagrama y la tabla de verdad del controlador de bus 8T97 (un controlador de bus equivalente es el 74367). Este dispositivo «triestado» es capaz de absorber 48 mA y puede admitir cualquier combinación de TTL, LSTTL y las conexiones de memorias que el usuario pueda desear realizar. La configuración final de buses de direcciones se muestra en la figura 4.11.

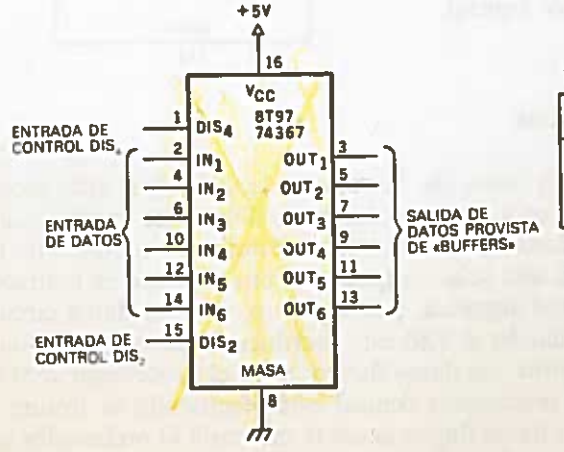


TABLA DE VERDAD

DIS_1	DIS_2	ENTRADA DE DATOS	SALIDA
0	0	0	0
0	0	1	1
X	1	X	Z ALTA
1	X	X	Z ALTA

X = «NO IMPORTA» (INDIFERENTE)
Z ALTA ES UNA CONDICION DE SALIDA DE «TRIESTADO»

Figura 4.10 *La salida de terminales y tabla de verdad de un controlador de bus 8T97/74367.*

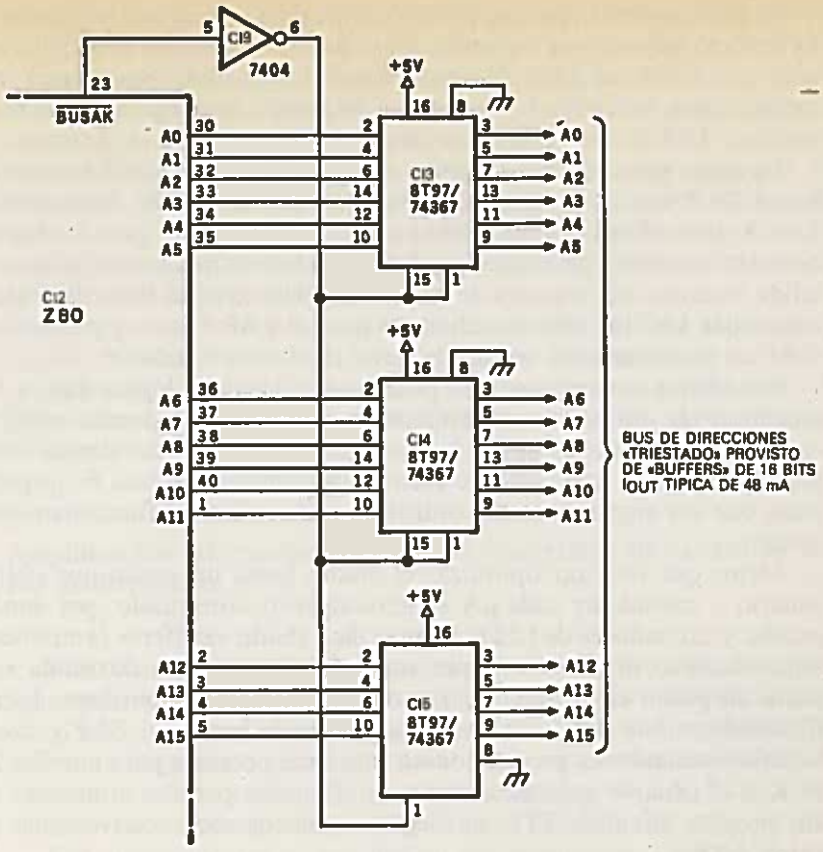


Figura 4.11 *La configuración final de bus de direcciones provista en «buffers».*

La función «triestado» del 8T97 es controlada por la señal $\overline{\text{BUSAK}}$. Esta señal conmuta el control del bus de direcciones a un dispositivo externo durante las operaciones de acceso directo a memoria (DMA). En una situación de «no DMA», $\overline{\text{BUSAK}}$ es de nivel alto y el 8T97 deja pasar todas las salidas procedentes del Z80. Cuando se acusa recibo de una petición DMA, $\overline{\text{BUSAK}}$ pasa a nivel bajo, poniendo el 8T97 en un modo de salida de alta impedancia. Estos medios permiten que la memoria sea escrita en, o leídas por, un dispositivo externo y suelen estar reservados para operaciones de alta velocidad que tienen más rapidez que la que puede conseguir el procesador central.

D. Bus de control y de datos

La cuarta y última zona de las conexiones directas del procesador central es el bus de datos y las restantes líneas del bus de control. La razón para proveer de «buffers» el bus de datos es similar al argumento para el bus de direcciones con una sola excepción: el bus de datos es bidireccional.

Un bus bidireccional significa, por supuesto, que los datos circulan en ambas direcciones. Cuando el Z80 está escribiendo un octeto de datos en una posición de memoria, los datos fluyen desde el procesador central a la memoria. Cuando el procesador central está efectuando la lectura de un octeto de memoria, los datos fluyen desde la memoria al procesador central. La naturaleza bidireccional del bus de datos exige que los controladores de buses sean bidireccionales internamente o estar unidos de tal forma que se realiza la misma función.

Una forma de hacer este controlador bidireccional es utilizar dos 8212s. El 8212 (figura 4.12) se concibió y produjo originalmente por Intel como un port de entrada o de salida, con enclavamiento, de 8 bits. El 8212 puede enclavarse continuamente de modo que los datos fluyan a través del mismo, o puede desconectarse para bloquear el flujo. Es idóneo para esta aplicación porque tiene una salida «triestado».

Dos circuitos 8212 (figura 4.13) están cableados en sentidos opuestos. CI6 dirige los datos desde el procesador central hacia la memoria, mientras que CI7 canaliza datos hacia el Z80. El control se ejerce a través de una sola línea conectada a la señal de control RD del procesador central. $\overline{RD}$ suele estar a nivel bajo salvo durante las operaciones de escritura. Ello hace que CI6 se desactive, en un modo «triestado», y CI 7 se active, lo que permite que los datos procedentes de la memoria o dispositivos de E/S lleguen al procesador central. Cuando RD adquiere el nivel alto durante una opera-

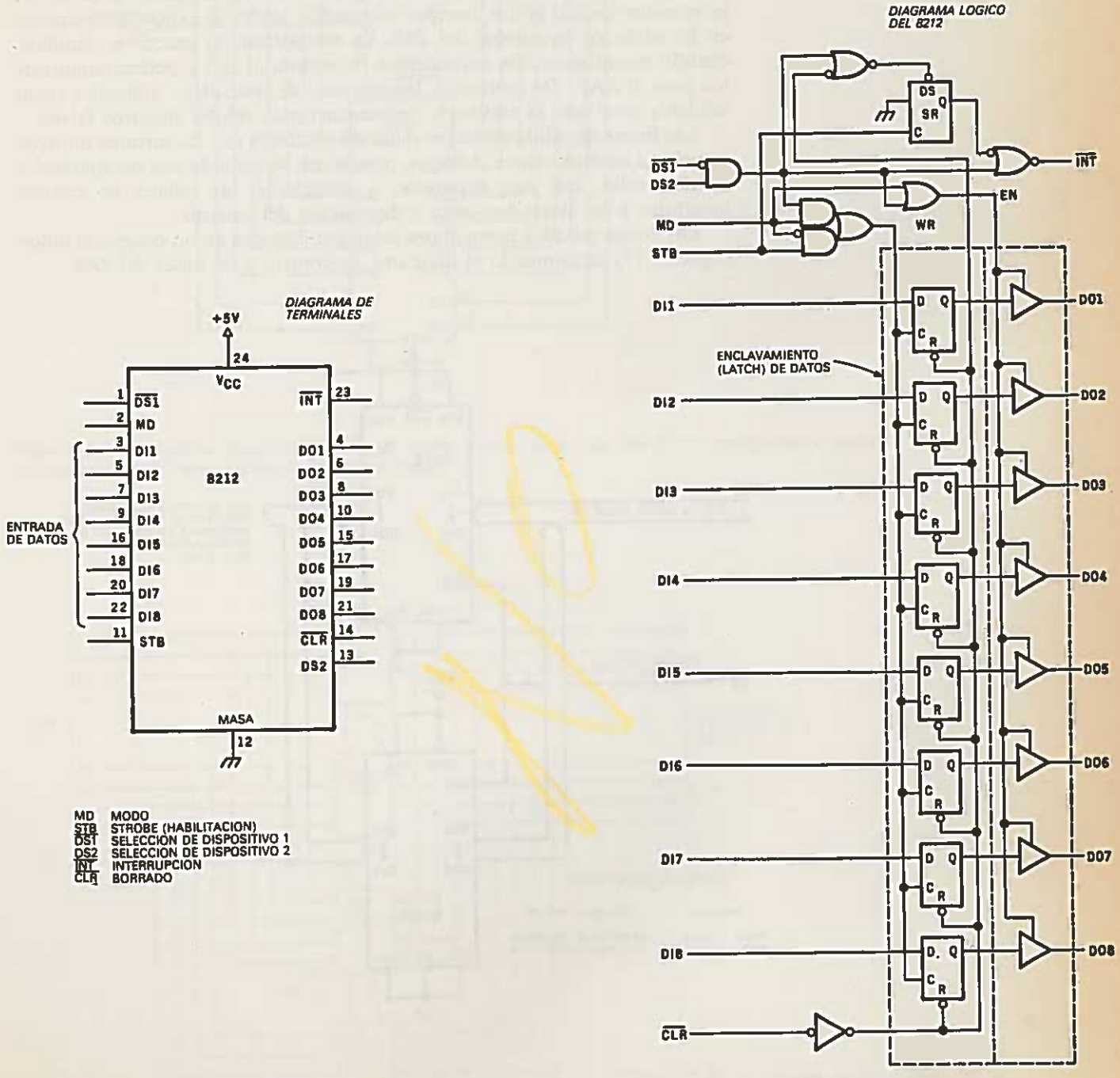


Figura 4.12 *La salida de terminales y diagrama lógico del port de E/S de 8 bits del 8212.*

ción de escritura, se invierte el proceso: CI 6 se activa e CI 7 se desactiva. Sólo es necesario utilizar la línea RD para controlar el sentido de los datos. Suponemos, naturalmente, que cuando el procesador central no está escribiendo datos, debe estar efectuando una lectura. Aunque no sea exactamente cierto, el concepto se aplica bastante bien en la práctica y los dos circuitos 8212 están esquemáticamente conectados como en la figura 4.14.

No es absolutamente necesario utilizar circuitos 8212 para realizar esta función. Los circuitos 8T97 ó 74367 también sirven para esta aplicación, pero precisan 4 cápsulas de circuitos integrados. Si no se tiene en cuenta el cableado adicional y se dispone de una fuente para los circuitos 8T97, pueden cablearse como se ilustra en la figura 4.15.

Las conexiones finales al procesador central a examinar son las señales del bus de control que se muestran en la figura 4.16. Coordinan los periféricos y los datos de canales y los direcciona hacia y desde el procesador central en los tiempos adecuados, según se explicó brevemente en la salida de terminales del Z80. La temporización exacta se detallará cuando examinemos los aditamentos de memoria, E/S y perfeccionamientos para el ZAP. De momento, las entradas de control no utilizadas están unidas a nivel alto (a través de resistencias) para inhibir disparos falsos.

Las líneas de salida están provistas de «buffers» por los mismos motivos que el bus de direcciones. Además, puesto que se trata de una computadora de desarrollo, con una expansión a considerar, las señales de control invertidas y no invertidas están a disposición del usuario.

Las zonas tratadas hasta ahora están combinadas en un diagrama único (figura 4.17) denominado el diagrama de control y de buses del Z80.

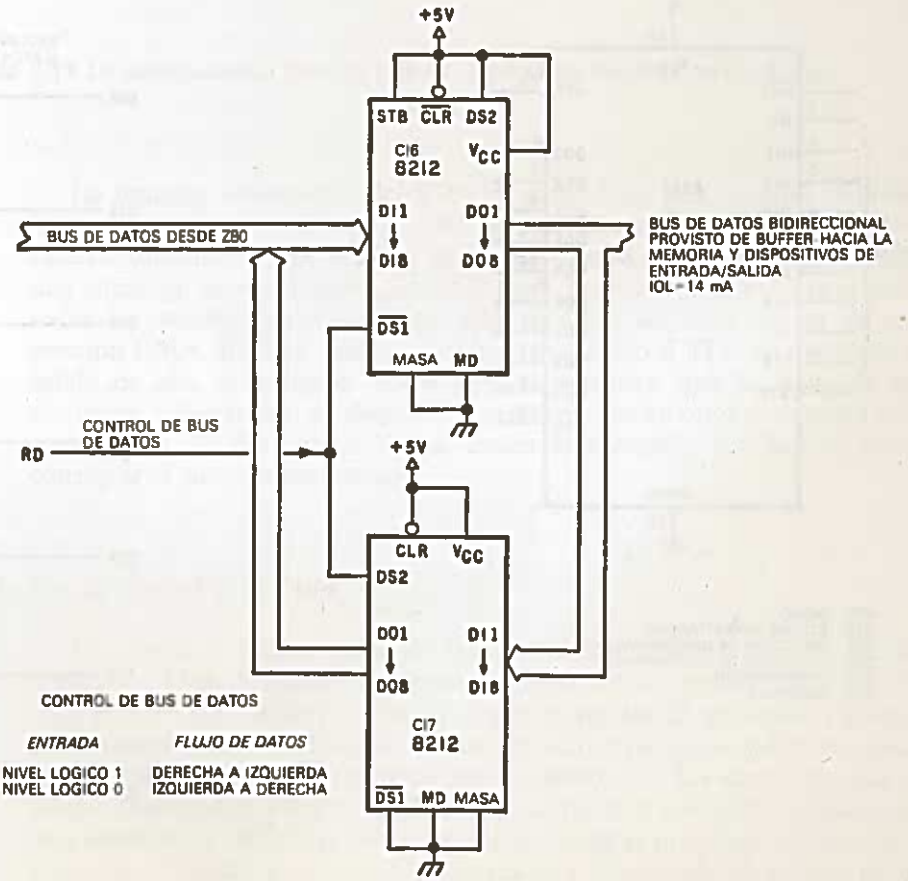


Figura 4.13 *Los circuitos 8212 configurados como controladores bidireccionales de buses de datos.*

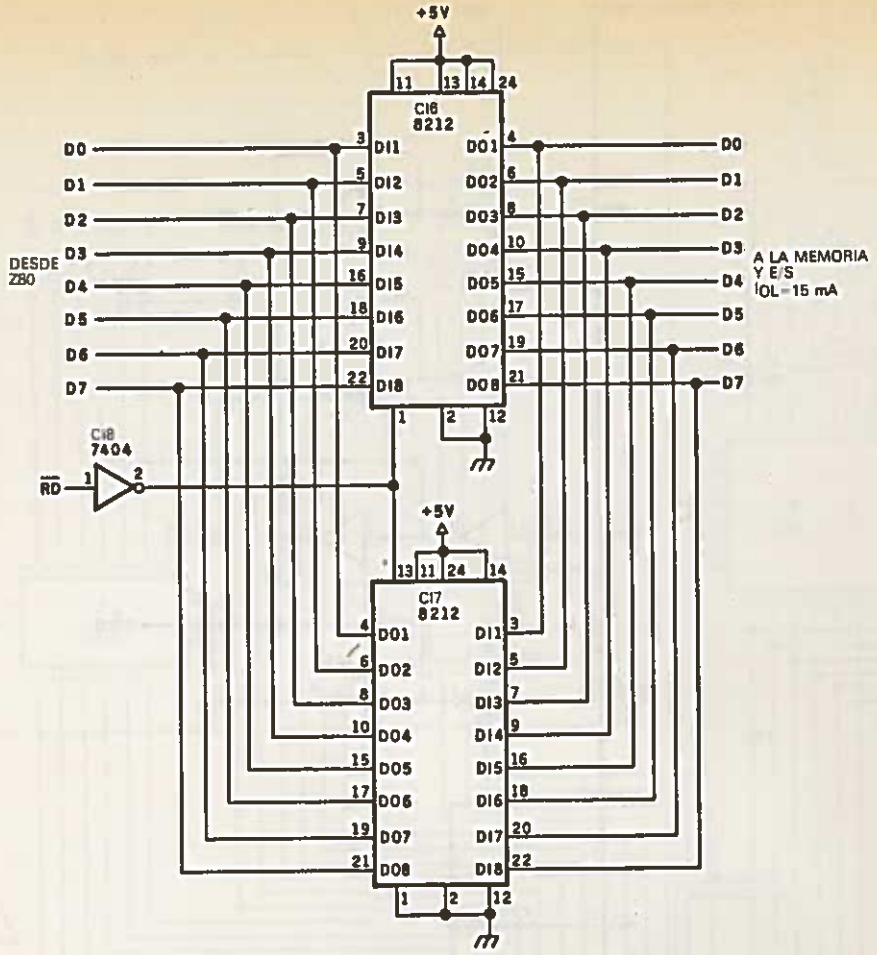


Figura 4.14 *Diagrama esquemático de los enclavadores de 8 bits del 8212 configurados como controladores bidireccionales de buses de datos.*

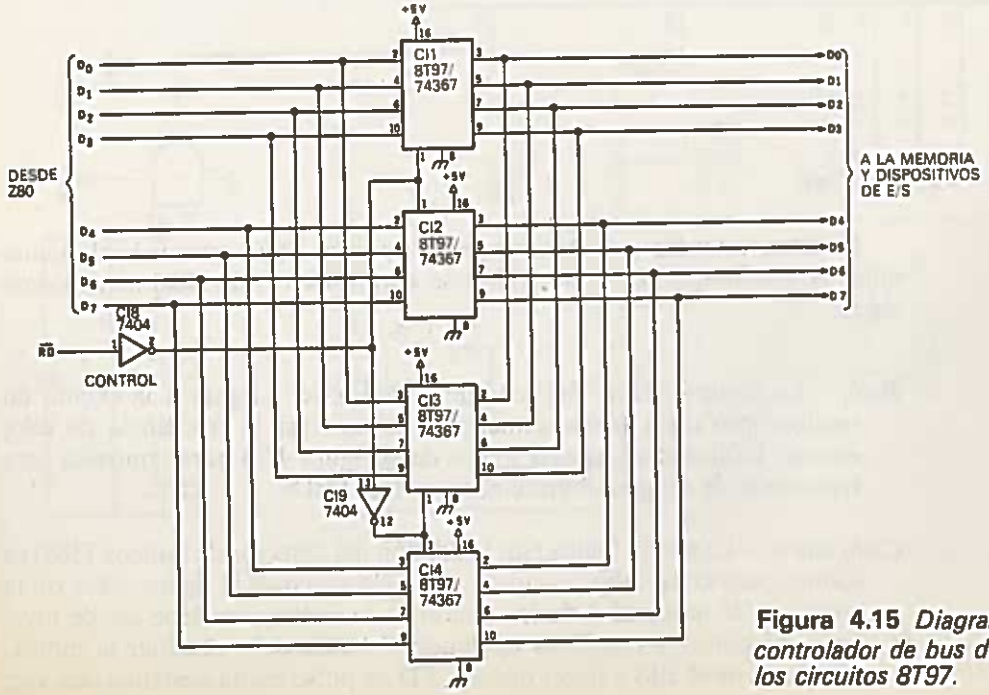


Figura 4.15 *Diagrama esquemático de un controlador de bus de datos configurado con los circuitos 8T97.*

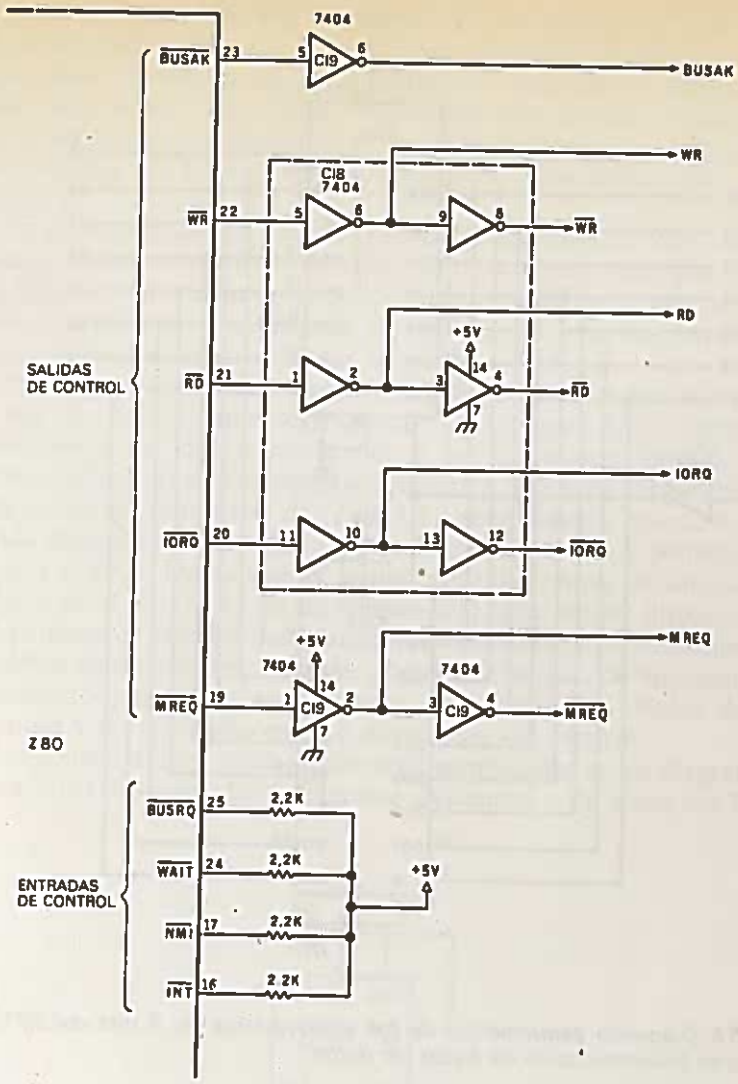


Figura 4.16 *Conexiones de entrada de control y salidas con aplicación de buffers del diseño de ZAP básico.*

E. Pruebas

Insertar todos los circuitos integrados salvo el Z80 y aplicar la alimentación. A continuación, cada sección se comprueba individualmente como sigue:

Reloj.—La comprobación del reloj de 2,5 MHz de la figura 4.3a exigirá un osciloscopio o un frecuencímetro para registrar la frecuencia de reloj exacta. Utilizando la sonda lógica de la figura 4.7b para controlar esta frecuencia de reloj se iluminarían los tres LEDs.

Ciclo único.—La sonda lógica (sin la adición del detector de flancos 7586) es idónea para comprobar el circuito de ciclo único de la figura 4.4. Con la sonda en el terminal 8 de la sección C, la indicación debe ser de nivel bajo. Al pulsar y mantener oprimido el botón debe cambiar la indicación a un nivel alto y hacer que el LED de pulso emita destellos una vez.

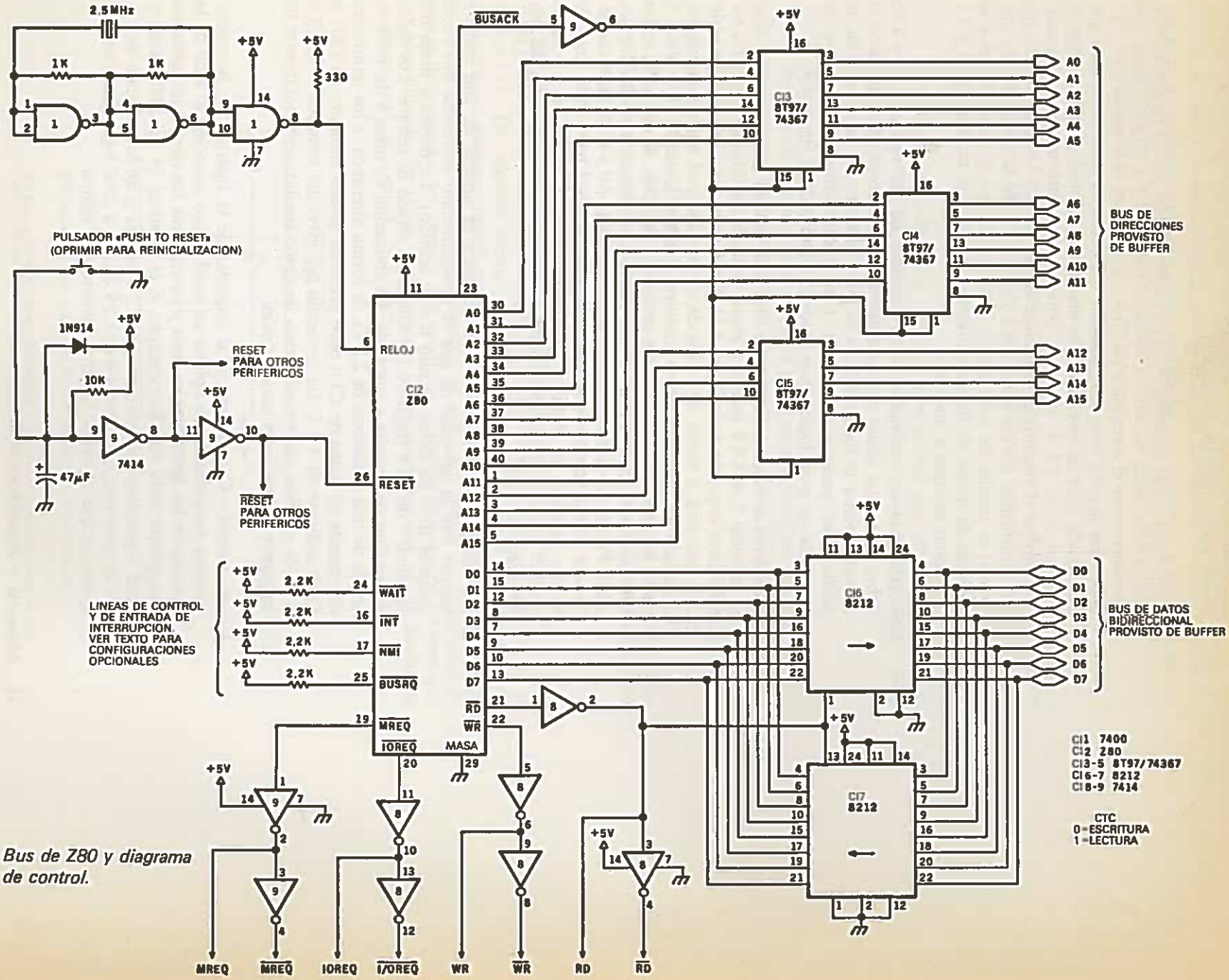


Figura 4.17 *Bus de Z80 y diagrama de control.*

Al soltar el botón no debe emitir destellos el indicador de pulsos, pues vuelve a su estado lógico inicial.

Paso único.—Con el conmutador en la posición del modo de paso único (figura 4.5), coger un cable conductor con pinzas y poner a masa *momentáneamente* el terminal 3 de CI 3. La salida en el terminal 8 de CI 1 debe estar al nivel bajo. La pulsación del botón de paso único hará que esta salida pase al nivel alto. Se mantendrá a nivel alto hasta que el terminal 3 de CI 3 vuelva a ponerse momentáneamente a masa. Comprobar el circuito de supresión de rebotes de los pulsadores (que está constituido por las secciones a y b de CI 1) de la misma forma que se hizo la prueba de ciclo único. Finalmente, con el conmutador a la posición de modo de funcionamiento continuo, el terminal 8 de CI 1 debe estar siempre a nivel alto.

Reinicialización del encendido.—Los circuitos de las figuras 4.8a y 4.8b deben tener una salida normalmente alta. Cuando se aplica primero la alimentación a la figura 4.8b, o se pulsa el botón en la figura 4.8a, la salida debe pasar a nivel bajo. Una u otra situación hará que se produzca un nivel lógico bajo desde el circuito de la figura 4.9.

Controladores de buses de direcciones.—El Z80 no debe estar insertado. Con el terminal 5 de CI 9 puesto a masa, todas las salidas de CI 3, CI 4 e CI 5 en el esquema de la figura 4.11 deben tener un nivel alto. En realidad éste será el modo de salida de «triestado» y el equipo de prueba adecuado les registrará como circuitos abiertos. Uniendo el terminal 5 de CI 9 a +5 V, a través de una resistencia de 2,2 K, se activarán todos los controladores de buses. Todas sus salidas estarán a niveles lógicos altos. Poniendo sucesivamente a masa las líneas A0 a A15, en el conector del Z80 debe obtenerse una indicación de nivel bajo en la línea de salida, provista de «buffers», correspondiente. Cuando las 16 líneas puedan hacerlo satisfactoriamente, se realiza la comprobación del bus de direcciones.

Bus de datos bidireccional.—El bus de datos se prueba de una manera similar con la salvedad de que el procedimiento se realiza dos veces (para el flujo de datos en uno u otro sentido). La puesta a masa del terminal 1 de CI 8 (figura 4.14) simula un estado de lectura. Los datos deben fluir de derecha a izquierda. La aplicación de masa y de +5 V (a través de una resistencia de 2,2 V), de forma alternada, a los terminales de entrada de datos de CI 6 debe producir niveles similares en DO1 a DO8 inclusive de CI 6. La elevación del nivel del terminal 1 de CI 8 a +5 V permite una transferencia de datos similar, pero solamente de izquierda a derecha en esta ocasión.

Bus de control.—Con referencia al esquema de la figura 4.16, la prueba consiste simplemente en aplicar un nivel lógico conocido al lado de la entrada de los inversores en serie y en registrar los niveles de salida en una puerta cada vez. Por ejemplo, si el terminal 19 del Z80 estaba a nivel lógico bajo, el terminal 2 de CI 9 estaría a un nivel lógico alto y, por el contrario, el terminal 4 de CI 9 estaría a nivel bajo. Cada sección inversora que atraviesa la señal invierte esta última.

II. Memoria y decodificación de E/S

Antes de que podamos usar dispositivos de E/S debemos aprender cómo actúa el direccionamiento del Z80. Recuerde que la dirección FF hexadecimal podría referirse a la memoria o a un port de entrada o de salida. La computadora debe tener la capacidad de diferenciar entre los tres posibles significados.

Las salidas de control del Z80 tienen la información necesaria para el encaminamiento, y seleccionando con puertas las señales de forma adecuada, se obtienen las señales correctas. Para operaciones de memoria y E/S básicas, las cuatro señales de especial interés son $\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$ y $\overline{WR}$. Sus definiciones son las siguientes:

A. $\overline{MREQ}$

Petición de memoria. Siempre que suceda una transacción entre el procesador central y la memoria, la línea $\overline{MREQ}$ pasa a nivel lógico «0».

B. $\overline{IORQ}$

Petición de entrada/salida. Siempre que suceda una transacción entre el procesador central y un port («puerto») de entrada o de salida, la línea $\overline{IORQ}$ pasa a un nivel lógico «0».

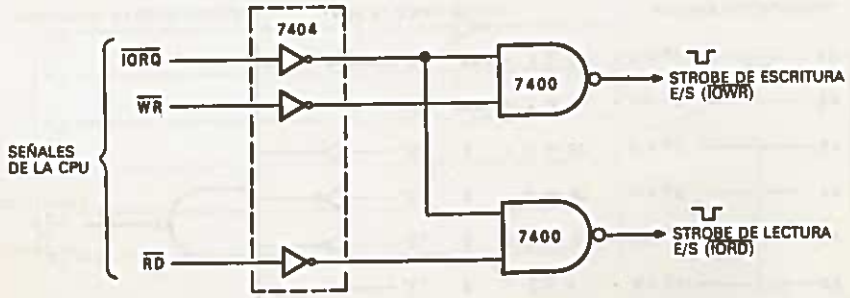
C. $\overline{RD}$

Petición de lectura. Siempre que el procesador central lea datos de entrada de la memoria o de un port de entrada, la línea $\overline{RD}$ pasa a un nivel lógico «0».

D. $\overline{WR}$

Petición de escritura. Siempre que el procesador central esté escribiendo datos en memoria o para un port de salida, la línea $\overline{WR}$ pasa a un nivel lógico «0».

Para diferenciar entre ports de entrada y salida durante las instrucciones de E/S, $\overline{IORQ}$, $\overline{RD}$ y $\overline{WR}$ se hacen pasar por las mismas puertas como se muestra en la figura 4.18. De manera similar, $\overline{MREQ}$, $\overline{RD}$ y $\overline{WR}$ pasan por puertas de conmutación durante transferencias de memoria, como se muestra en la figura 4.19. A diferencia con la decodificación de E/S, pero similar al controlador de bus de direcciones antes tratado, una condición de lectura de memoria no tiene que ser decodificada. Se supone que cuando la memoria no está en un modo de escritura, está en el estado de lectura.



IORQ PASA A UN NIVEL LOGICO «0» EN UNA OPERACION DE ENTRADA/SALIDA

WR PASA A UN NIVEL LOGICO «0» CUANDO LA CPU INTENTA ESCRIBIR DATOS EN UNA SALIDA O EN UNA MEMORIA

RD PASA A UN NIVEL LOGICO «0» CUANDO CPU INTENTA LEER DATOS DESDE LA MEMORIA O UN DISPOSITIVO DE ENTRADA

Figura 4.18 *Decodificación de lectura y escritura de entrada/salida.*

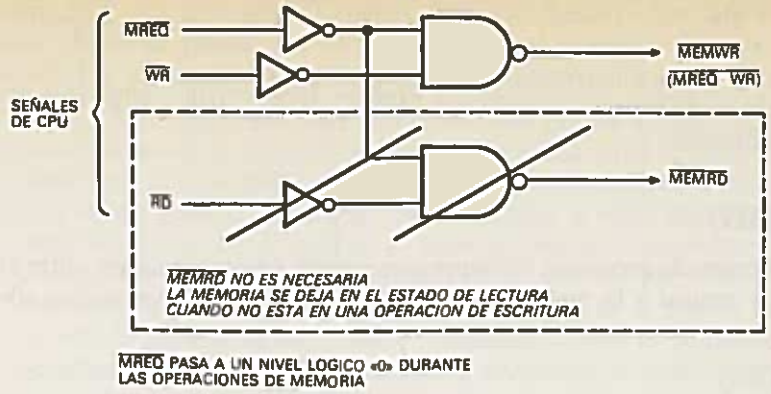


Figura 4.19 *Decodificación de lectura y escritura de memoria.*

Las tres señales «strobe» decodificadas resultantes definen las operaciones de lectura del port de entrada ($\overline{\text{IORD}}$), escritura de port de salida ($\overline{\text{IOWR}}$) y escritura de memoria ($\overline{\text{MEMWR}}$). Si solamente fueran requeridas tres funciones en su configuración particular de computadora, entonces no sería necesario ninguna otra decodificación. Tal computadora tendría un port de entrada, un port de salida y un banco de memoria. Para solventar este problema, la decodificación adicional de E/S y la memoria es necesaria de forma que estas señales strobe puedan servir a más de un solo dispositivo. Con la circuitería adicional, el Z80 puede direccionar independientemente 256 ports de entrada y salida y 64 koctetos de memoria.

Durante una petición de E/S (entrada o salida), la dirección binaria de 8 bits del port particular de E/S aparece en las líneas de A0 a A7 inclusive del bus de direcciones. Una explicación de la codificación de direcciones se muestra en la figura 4.20. Ejemplos adicionales se ilustran en la figura 4.21.

Usando esa información, si una instrucción hubiera de designar el port de salida 7 como su destino, entonces la circuitería de la figura 4.22 podría ser utilizada. Cuando un código de 007 octal (07 hexadecimal o 00000111 en binario) aparece en las líneas de dirección con una señal strobe $\overline{\text{IOWR}}$, las señales presentes en el bus de datos serían almacenadas en un registro de 8 bits como datos de salida.

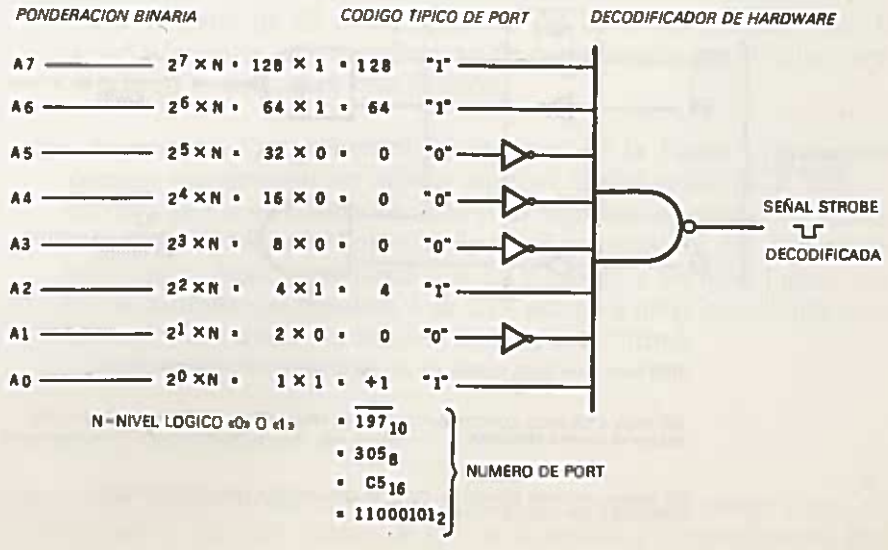


Figura 4.20 *Una explicación de códigos de dirección de entrada/salida.*

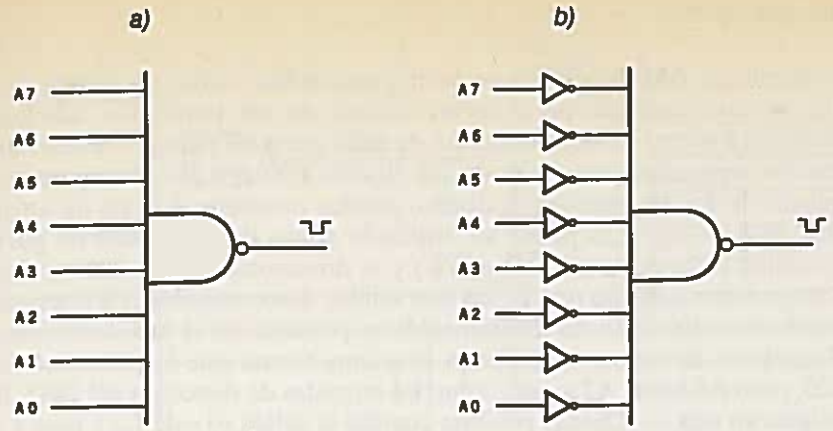


Figura 4.21 *Lógica de decodificación de dirección.*
a) Para dirección FF_{16}.
b) Para dirección 00_{16}.

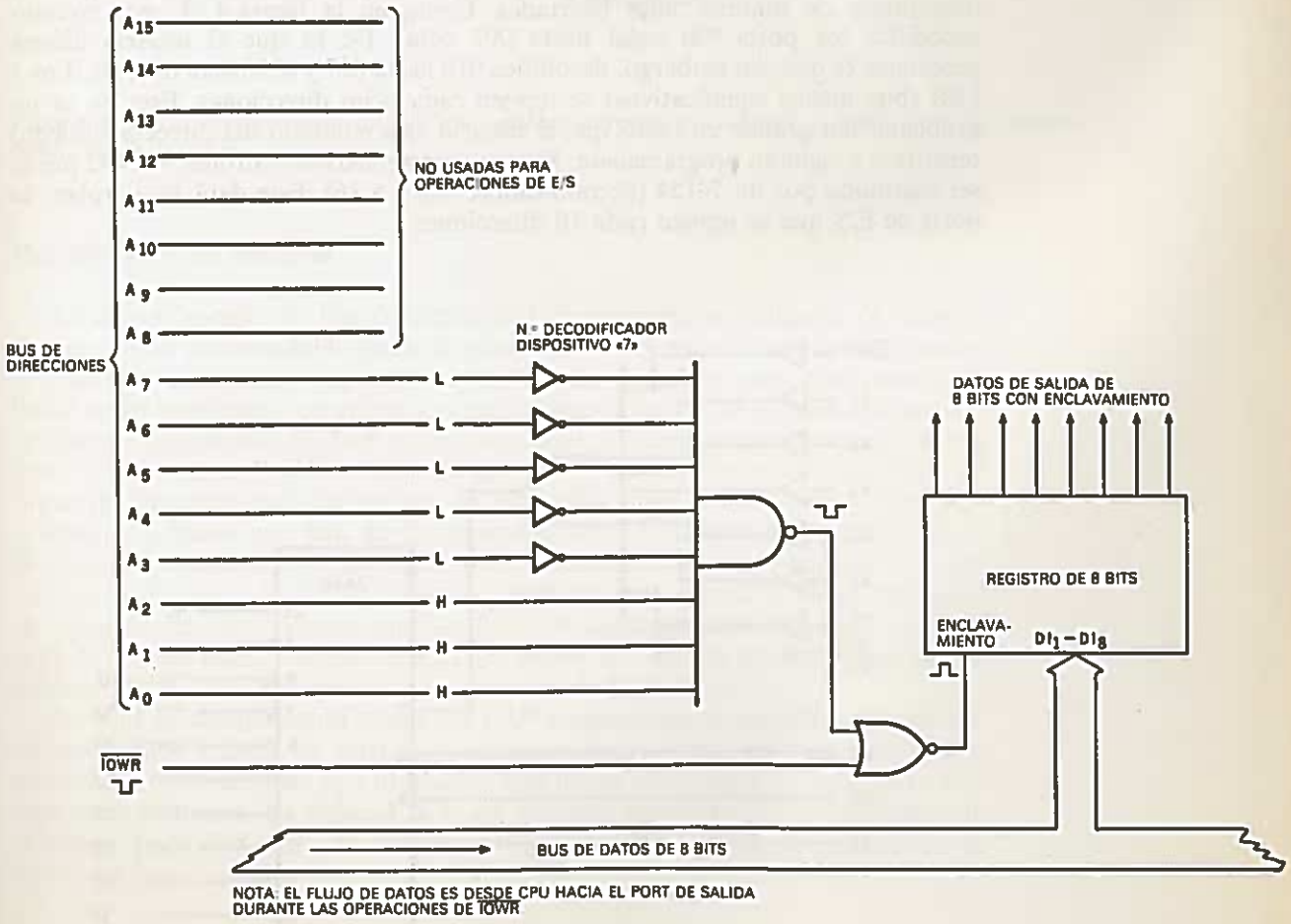


Figura 4.22 *Un método posible para decodificar una dirección única de port de salida de 8 bits. El circuito es para un código de dispositivo* 007_8.

Decodificación de E/S

Por supuesto, el ZAP necesita más de un port, incluso como un sistema básico. De hecho, si es expandido para incluir alguno de los periféricos adicionales, requerirá de 6 a 8 ports. La decodificación de estos ports no requiere necesariamente 8 circuitos separados como los de las figuras 4.20 ó 4.21. Incorporando un demultiplexor de 4 a 10 líneas en el diseño, pueden obtenerse 8 ports de «strobe». El circuito de la figura 4.23 puede ser utilizado como decodificador de ports de entrada o salida (seleccionando $\overline{RD}$ o $\overline{WR}$) y es direccionado para 000 octal a 007 octal. Trabaja seleccionando una de las dos salidas desconectadas (CI 3 terminal 9 o 10) cuando una dirección no decodificable se presenta en el bus de direcciones. A3 a A7 inclusive deben ser tratadas en la misma forma que la presentada en la figura 4.20, pero A0 hasta A2 sirven como las entradas de dirección del 7442. Estos 3 bits designarán una de 8 líneas posibles cuando la salida «1» de CI 1 pase a nivel bajo.

La duplicación de este circuito para proporcionar 8 señales strobe separadas de entrada y salida (direccionada de 000 hasta 007) requeriría un total de 7 chips. El número de chips puede ser reducido a 3 si tomamos una pequeña licencia poética en el diseño. Hasta aquí hemos decodificado los 8 bits de la sección de E/S del bus de direcciones haciendo seleccionar a nuestro decodificador 1 de 256, o, como en el circuito anterior, 8 de 256. En uno u otro caso, sólo las direcciones designadas tienen alguna importancia. Las demás carecen de significado. Para fines prácticos podríamos decodificar las líneas A0 hasta A2 e ignorar el resto. Un circuito que hace esta función se muestra en la figura 4.24.

La diferencia entre este circuito y los previamente descritos, además de tener menos chips, es que éste requiere de alguien que reconozca las ventajas y desventajas de tomarse tales libertades. Como en la figura 4.23, este circuito decodifica los ports 000 octal hasta 007 octal. De lo que el usuario deberá percatarse es que, sin embargo, decodifica 010 hasta 017 y 020 hasta 027, etc. Los 3 LSB (bits menos significativos) se repiten cada ocho direcciones. Este no es un problema tan grande en tanto que el usuario está enterado del direccionamiento repetitivo y vigile su programación. Si se precisaran más de 8 strobes, el 7442 puede ser sustituido por un 74154 (decodificadores de 4 a 16). Este dará 16 «strobe» de ports de E/S que se repiten cada 16 direcciones.

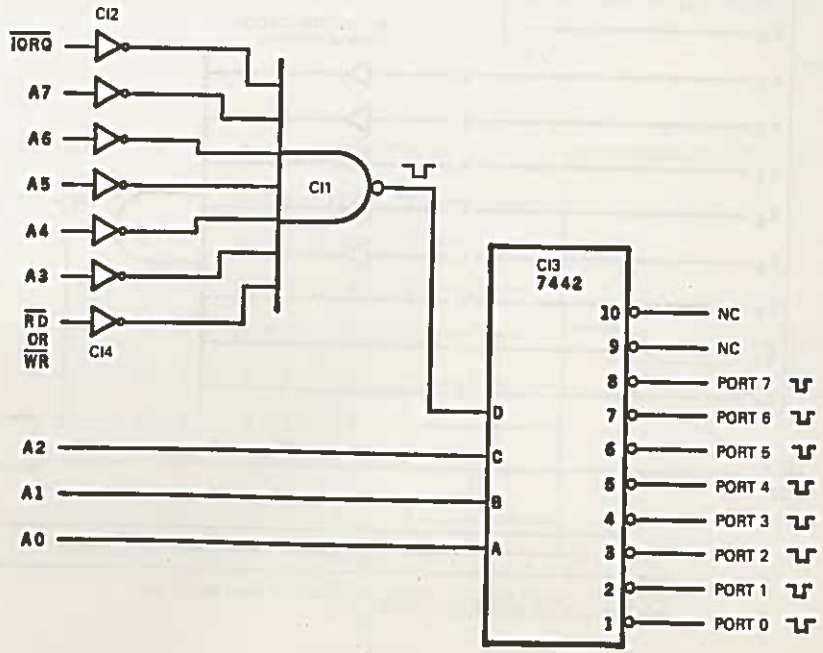


Figura 4.23 *Un método formal de decodificación de direcciones de ports de entrada y salida que decodifican las 8 líneas de dirección.*

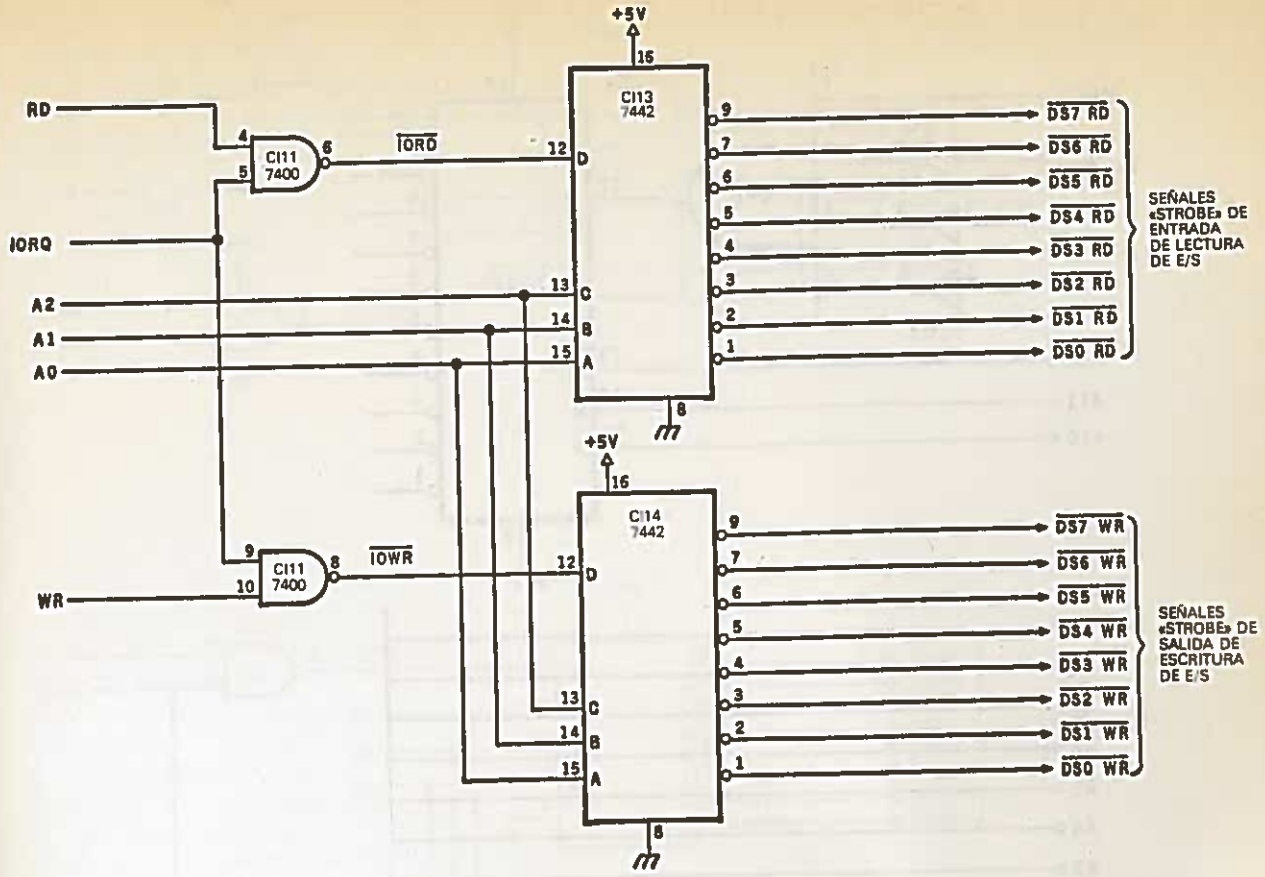


Figura 4.24 *Un método para decodificar señales «strobe» de entrada y salida con un número reducido de circuitos.*

Decodificación de memoria

La decodificación del bus de direcciones de memoria es realizada de manera similar. No es recomendable seguir el mismo camino y permitir el direccionamiento repetitivo de la memoria porque hay más probabilidad de error. Aun cuando 16 líneas están implicadas, en aplicación real la decodificación de la memoria viene a ser menos complicada. El ZAP utiliza bancos de memoria programable de 1 K × 8 bits y una memoria ROM borrable de 1 kocteto. Ambos dispositivos requieren 10 líneas de dirección para definir una de 1024 posiciones en cada banco. Ello deja solamente 6 líneas que han de decodificarse individualmente para definir algún bloque de memoria de 1 K. La figura 4.25 ilustra cómo se puede realizar esto. Un 7442 (decodificador de 4 a 10 líneas) es utilizado para generar 8 líneas separadas de selección de chips. Puesto que las líneas de dirección del 7442 están unidas a A10-A12, cada pulso «strobe» tendrá un límite de 1 K. No es casualidad que se eligiera 1 K×8 como la capacidad de memoria de cada banco.

Aunque la configuración básica del ZAP proporciona la decodificación de 8 K de memoria y 8 ports de entrada y salida, no todas estas selecciones de chip y señales «strobe» de ports son utilizadas. Las líneas adicionales se dejan para una expansión posterior. La figura 4.26 es un esquema completo del decodificador de memoria y de E/S para el constructor que ha de añadirse al circuito de la figura 4.17.

Pruebas

Después de que se hayan añadido los componentes de la figura 4.26 a la figura 4.17, se estará en condiciones de probar la decodificación de E/S y memoria.

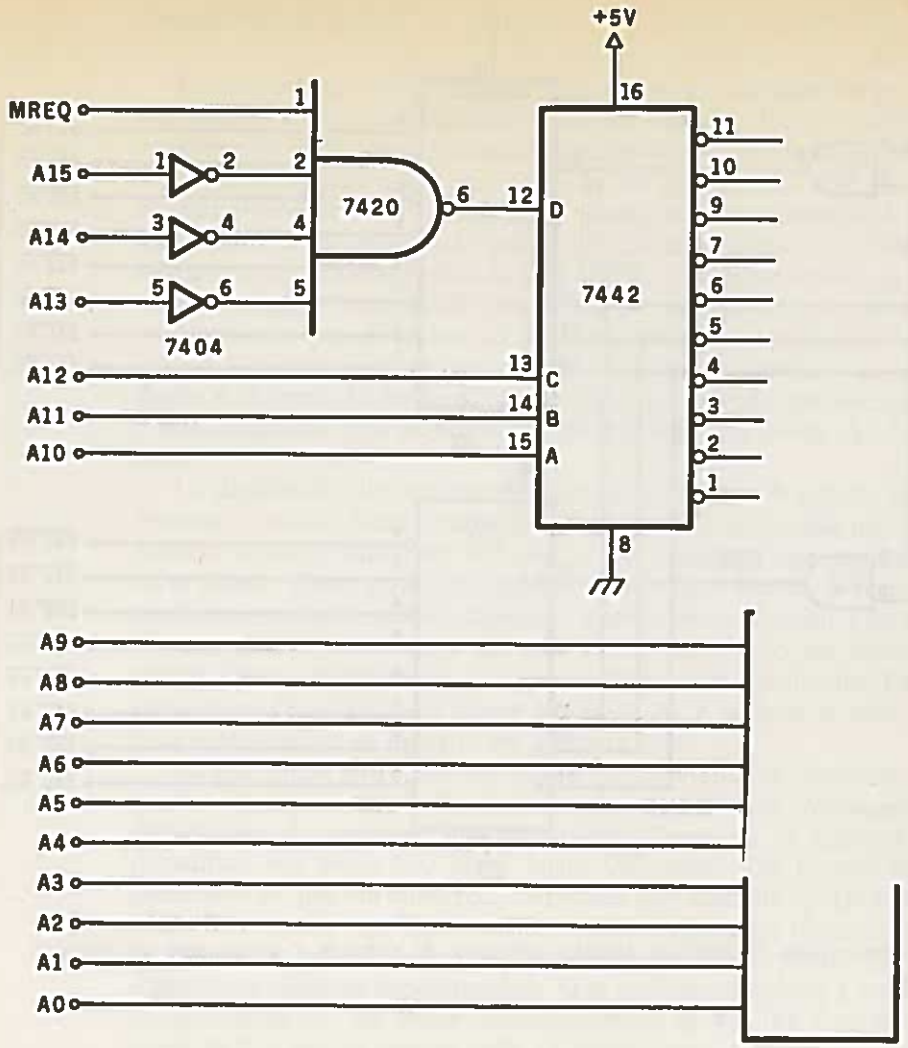


Figura 4.25 *Decodificación de banco de memoria para 8K de memoria.*

Inserte los circuitos integrados CI 10, 11, 12, 13 y 14, pero no inserte CI 20 todavía. Los CI 1, 3 y 9 deben permanecer en su posición de la prueba previa. El Z80 deberá dejarse fuera todavía. El nivel lógico en la entrada de dirección D de cada uno de los 7442 (CI 12, 13 y 14) debe estar alto. Eliminando los CI 8 y 9 (con el sistema *apagado*) se producirá en esta entrada un cambio inmediato a un nivel lógico bajo. CI Seguidamente, se ponen a masa los terminales 30, 31 y 32, y 23 a nivel alto, en el zócalo del Z80. Con los buffers del bus de direcciones activados, y una dirección 000 puenteada en A0 hasta A2, un nivel bajo de selección de chips debe aparecer en la dirección de «strobe» más baja. En este caso, el terminal 1 de los circuitos CI 13 y 14 deberá estar a nivel bajo y las otras líneas de «strobe» a nivel alto. Cambiando los tres puentes de A0 hasta A2 se permitirán otros «strobes» de selección de chips de dispositivos. El decodificador del banco de memoria trabaja de la misma forma, excepto que el puente debe ser aplicado a las líneas de dirección A10 hasta A12.

Después de la prueba, inserte todos los chips excepto el de Z80.

III. Memoria

Naturalmente, se trata de la parte más importante de cualquier sistema de computadora. Tanto las instrucciones de programación como los datos deben ser almacenados en el tiempo adecuado de tal forma que la computadora pueda realizar su función. Aun cuando el procesador central del Z80 tiene una cantidad

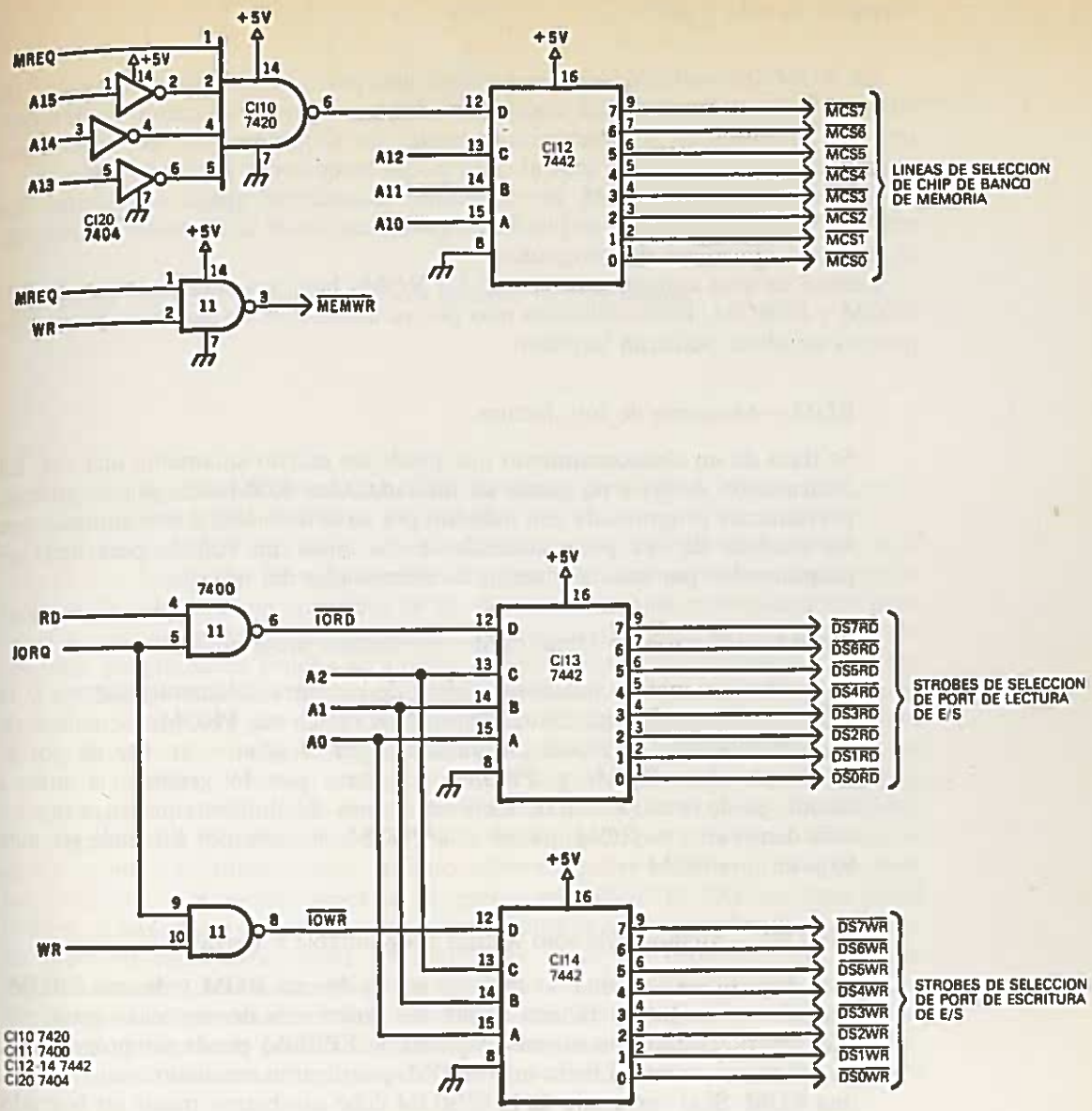


Figura 4.26 *La sección de decodificación de entrada y salida y de memoria del ZAP.*
a) Strobes de selección de chip del banco de memoria.
b) Strobes de selección de chips de dispositivos de E/S.

de registros de almacenamiento de 8 bits, éstos pueden ser utilizados solamente en la manipulación temporal de datos y no pueden almacenar instrucciones de programas. Las instrucciones de programación deben ser almacenadas en elementos de memoria exterior.

La memoria externa puede ser dividida en dos amplias clases: ROM (memoria de sólo lectura) y RWN (memoria de lectura/escritura). La ROM es utilizada para almacenar específicos, pasos permanentes de programación o datos. Los contenidos de estas posiciones de memoria son considerados permanentes y no pueden cambiarse fácilmente. La memoria de lectura/escritura, por otra parte, es utilizada para almacenar datos que cambian cuando la computadora está funcionando. Podrían ser ejemplos los resultados de cálculos o programas que varían frecuentemente. Para uno u otro tipo de memoria, la última función permanece igual: proporcionar, al solicitárselo, o una instrucción para ejecución o una posición en donde los datos puedan ser almacenados.

Memoria de sólo lectura

La ROM (memoria de sólo lectura) es una parte importante del sistema de computadora. ROM funciona como una matriz de memorias cuyos contenidos, una vez establecidos por técnicas especiales de programación, no pueden ser alterados por el procesador central. Hay pocas excepciones para esta regla.

Por su naturaleza, ROM no es volátil. Cuando se apaga el sistema, los contenidos del programa no se pierden. La reaplicación de la alimentación permite la ejecución inmediata del programa.

Dentro de esta categoría básica de las ROMs hay tres subcategorías: ROM, PROM y EPROM. Están definidas más por su utilización y aplicación que por lo que sus nombres pudieran implicar.

ROM.—Memoria de sólo lectura

Se trata de un almacenamiento que puede ser escrito solamente una vez. La información es fija y no puede ser alterada. Una ROM está, por lo general, previamente programada con máscara por su constructor y es comprado con un modelo de bit preestablecido. Estos tipos de ROMs permiten ser programados por encargo, según las necesidades del usuario.

PROM.—(Usuario).—Memoria de sólo lectura programable

Este almacenamiento puede ser objeto de escritura solamente una vez y la información queda fija. Estos dispositivos suelen ser PROMs de enlace de fusibles bipolares, que son programados por el usuario en vez de por el fabricante. Las ROMs y PROMs no usan, por lo general, la misma tecnología de construcción de semiconductores. El almacenamiento es mucho más denso en una ROM que en una PROM, el coste por bit suele ser más bajo en una ROM.

EPROM.—Memoria de sólo lectura programable y borrable

Este dispositivo combina las mejores partes de una ROM y de una PROM. Cuando se recibe de fábrica todas las posiciones de memoria están sin programar. Usando un interfaz especial, la EPROM puede ser programada por el usuario como si fuera una PROM, y utilizar el resultado como si fuera una ROM. Si el contenido de la EPROM debe cambiarse, puede ser borrado y programado de nuevo. Dependiendo del dispositivo particular que se pueda añadir, una EPROM puede ser cambiada electrónicamente (a menudo diferenciado por la abreviatura EAROM) o borrable por radiación ultravioleta. Esta última se denomina a veces UVEPROM, pero es más frecuente sólo llamarla EPROM. Son fácilmente reconocibles porque tienen una ventana de cuarzo sobre el circuito integrado. Esta ventana es transparente a la luz ultravioleta y facilita el borrado.

En tanto que se pueden discutir los méritos de cada opción, todas las ROMs realizan la misma función final. Para cada posición direccionable, independientemente hay una configuración específica de bits almacenados. Sólo el procesador puede determinar si se trata de datos o de una instrucción. En cualquier caso el método de almacenamiento es el mismo. En la figura 4.27 se detalla el diagrama de bloques de una ROM.

Una ROM es simplemente un bloque lógico que, bajo el control del programa, proporciona una configuración preestablecida. La figura 4.28 es una memoria de lectura de 3 bits. Cuando el interruptor SW1 se cierra (la posición se adoptaría cuando el procesador central quisiera la información almacenada), el código de 3 bits de «101» aparecería en las salidas. El diodo pone a tierra las señales de entrada para los inversores 7404, cuando SW1 está cerrado. La ampliación a más de 3 bits implica simplemente utilizar más diodos, resistencias y etapas de buffers. Tal

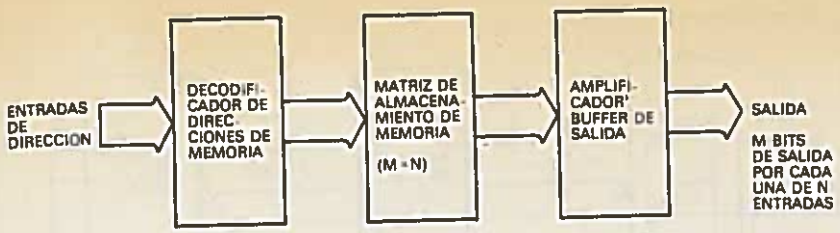


Figura 4.27 *Diagrama de bloques de una memoria de sólo lectura.*

circuito está considerado como una ROM de matriz de diodos y, en este caso, sería considerado una ROM de 1 línea por n-bits.

Una memoria de 3 bits no es muy usada. Este sistema puede ser ampliado a 16 octetos con la adición de un decodificador de direcciones como se ilustra en la figura 4.29. Un esquema completo de los diodos dispuestos específicamente para hacer funcionar un simple programa de 9 octetos se ilustra en la figura 4.30. Este pequeño programa de prueba será utilizado después durante la fase de revisión.

La ROM de matriz de diodos es presentada únicamente por un valor educativo. Este no es un método que deba ser empleado en la computadora ZAP. Conscientes de que hay circuitos integrados que podrían satisfacer los requisitos en cada una de las tres categorías, debemos analizar nuestras necesidades un poco más de cerca.

Las cuestiones pertinentes son: Capacidad de memoria y el coste y la facilidad de programación. El tamaño de la ROM es determinado por el usuario. Cuando se aplica primero la alimentación, ¿cuánto esfuerzo quiere realizar el usuario para hacer ejecutar a la computadora un programa específico? El ZAP no tiene panel frontal, ni bancos de direcciones, ni conmutadores de datos para las instrucciones. Al estar en estas condiciones, el ZAP debe tener un programa que ejecute inmediatamente (cuando se aplique la alimentación o se pulse el botón de reset) y que permita al procesador central comunicarse con sus periféricos y ponerse, asimismo, en un modo que sea directamente programable mediante estos dispositivos. Una vez aplicada la alimentación, puede escribirse un simple programa de 50

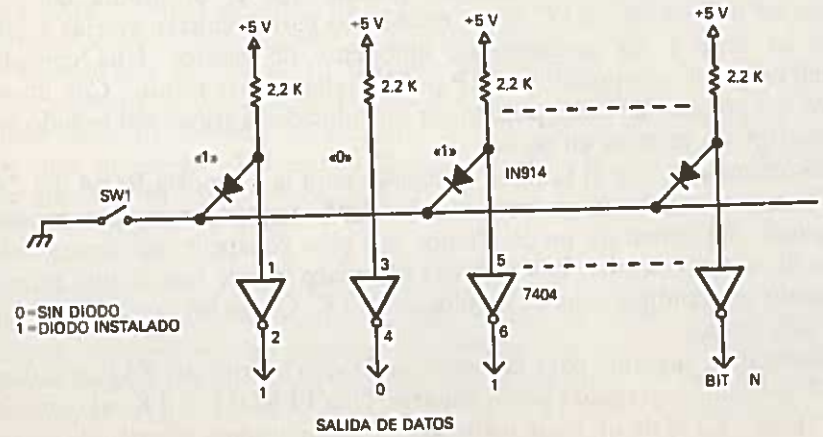


Figura 4.28 *Una memoria simple de sólo lectura de 3 bits (1 × 3 bits).*

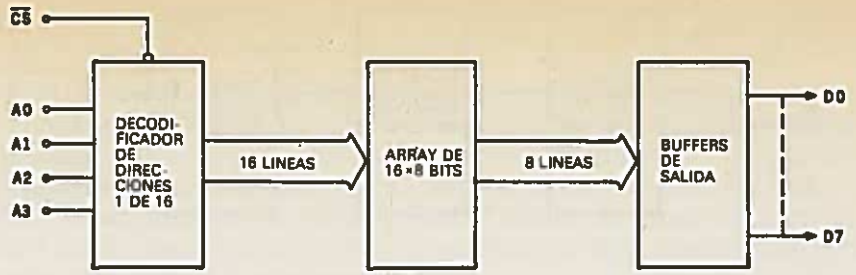


Figura 4.29 *Un diagrama de bloques de una memoria de sólo lectura de 16 octetos.*

a 100 octetos, que facilita el teclado para la carga de memoria. Pero quizá necesitemos introducir un programa grande en memoria. ¿Hemos de introducirlo en su totalidad mediante el teclado?

La entrada de datos de alta velocidad puede conseguirse a través de interfaz en serie. Este puede añadirse a expensas de otros 100 ó 200 octetos. Otra consideración es la necesidad de alguna visualización de datos y direcciones del operador para facilitar el desarrollo del programa.

En conclusión, para incorporar todas las funciones necesarias para un sistema de desarrollo de una sola placa, la ROM puede requerir fácilmente de 500 a 1.000 octetos de almacenamiento. Muchos sistemas de computadoras usan una ROM de 64 a 256 octetos para almacenar un programa de autocarga («bootstrap»), que es un programa que coordina la menor cantidad posible de periféricos necesarios para cargar un programa más grande en la computadora. En la mayoría de los sistemas de computadoras personales, este sistema controla un interfaz de casete y el programa que es almacenado posteriormente es llamado monitor.

Un monitor (explicado en el Capítulo 6) es una pieza muy importante de software que necesita aproximadamente un 1 K de almacenaje de programa. Nuestra decisión está en hacer el monitor totalmente residente en la ROM (lista para ejecución inmediata), o en reducir la ROM al mínimo y cargar el monitor desde un teclado o a partir de un sistema de almacenamiento de casete.

Esta es una consideración importante para alguien que esté construyendo una computadora desde sus comienzos. Cuando se da una opción a escoger, considero que casi siempre deberá optar por la solución de los menos componentes posibles y deberá incluir el monitor de ROM en el hardware. Es como poner el carro antes que el caballo, pretender que un interfaz de casete sea usado para cargar todo el software de diagnóstico. Es bastante posible que el programa del monitor, residente en una ROM de 1 K, fuera necesitado para localizar averías y alinear el interfaz en serie y las secciones de «modem» de casetes. Una consideración adicional es que la computadora ZAP puede trabajar más pronto. Con un monitor de ROM los programas útiles pueden ser introducidos a través del teclado sin tener que construir un interfaz en serie.

Es recomendable que el tamaño adecuado para la memoria ROM del ZAP sea de 1 K. Como se mencionó previamente, la ROM se preprograma con máscara por el fabricante. Sin embargo, no olvidemos que para construir una computadora en casa, usted es el fabricante. Las PROMs de enlace fusible son de una proposición cara cuando son configuradas en un bloque de 1 K. Como un cargador inicial de 64 octetos son ideales.

La alternativa sugerida para la memoria de sólo lectura del ZAP es utilizar una EPROM que sea programada por el usuario. Una EPROM de 1 K tal como la 2708 (o la 2716 de 2 K) es de un coste justo para la computadora construida en casa. La memoria de sólo lectura, borrable a través de radiación ultravioleta, 2708, es recomendada para esta aplicación. (La 2716 es una EPROM de 2 K con una fuente única de poder +5 V.)

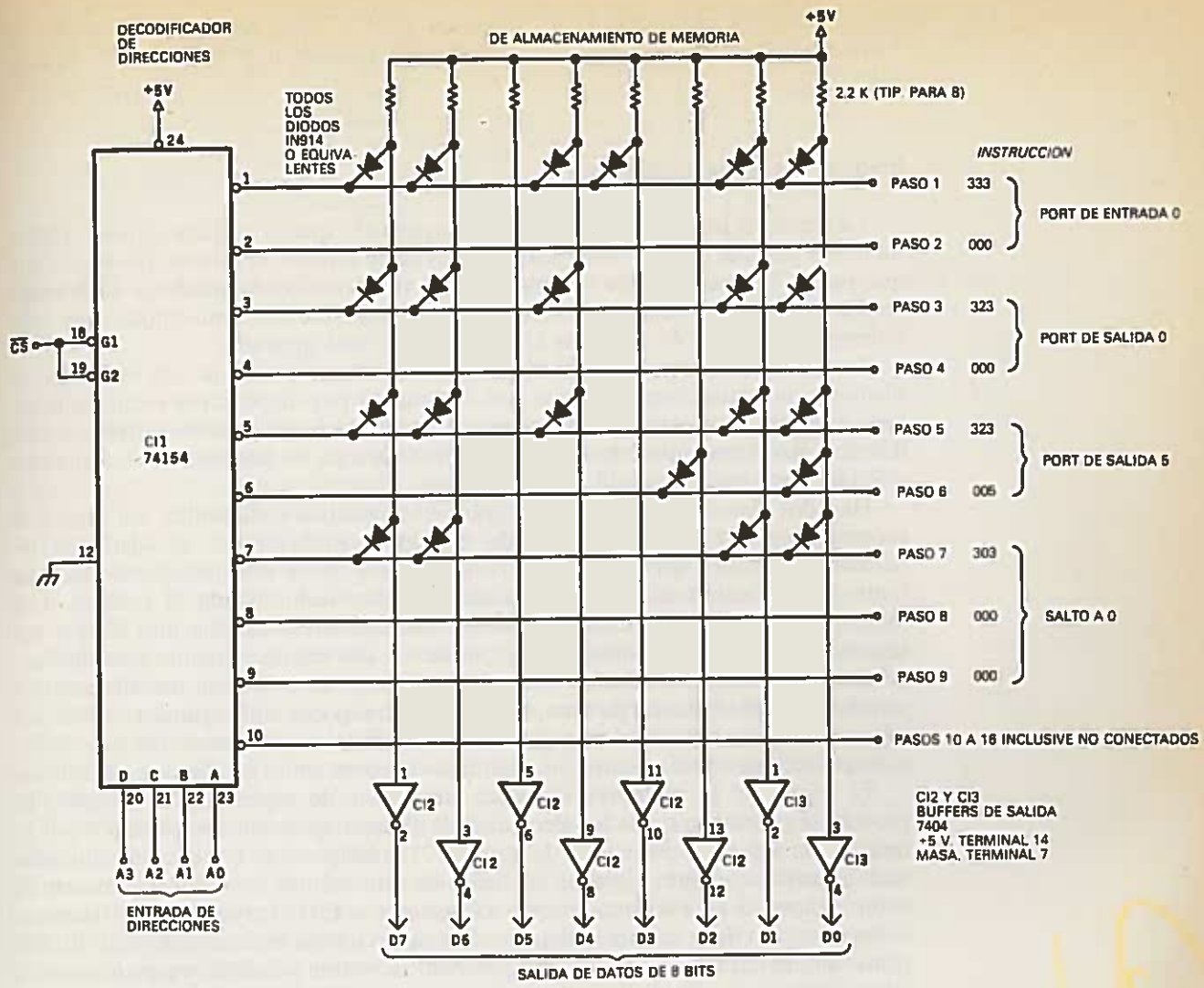


Figura 4.30 *Una memoria de sólo lectura de matriz de diodos con un programa de prueba.*

Las memorias EPROMs

La EPROM es una memoria de lectura en su mayor parte. Es usada como una ROM para periodos prolongados de tiempo, borrada ocasionalmente y reprogramada cuando sea necesario. El borrado es realizado por la exposición de un pequeño sustrato de chip, cubierto con una ventana transparente de cuarzo, a la luz ultravioleta. El elemento de memoria de EPROM usado por Intel en el 2708 es de un tipo de carga almacenada llamado transistor FAMOS (dispositivo de almacenamiento MOS de inyección por avalancha y puerta flotante). Es similar a un FET de puerta de silicio y canal p, con la puerta «flotante», o inferior, completamente rodeada por un aislador de dióxido de silicio. El valor de almacenamiento de «1» ó «0» del elemento FAMOS es una función de la carga en la puerta flotante. Un elemento cargado tendrá la salida de almacenamiento opuesta de un elemento no cargado. Al aplicar una tensión de carga de 25 V a elementos selectivamente direccionados, configuraciones de bits particulares que constituyen el programa, pueden escribirse en la EPROM. Rodeada por material aislante, la carga puede durar años. Cuando este aislador de dióxido de silicio se expone a una intensa luz ultravioleta, se hace algo conductor y purga la carga en la puerta flotante. El resultado es el borrado de toda la información programada.

En los Apéndices C1 y C2 se detallan la disposición de los terminales y las

especificaciones eléctricas de los circuitos 2708 y 2716, respectivamente. En el Capítulo 7 se examinan diversos métodos para programar y probar la pastilla integrada.

Memoria de lectura/escritura

La memoria de lectura/escritura es justamente lo que su nombre implica. Dicha memoria permite que los datos sean escritos en su interior lo mismo que se leen los que posee. La memoria de lectura/escritura para microcomputadoras suele estar configurada por dispositivos de memoria programables de semiconductores, que retienen los datos sólo mientras la alimentación está aplicada.

Las ROM son técnicamente dispositivos de acceso aleatorio; sin embargo, la memoria de lectura/escritura, que está compuesta por dispositivos semiconductores y está primeramente concebida para su uso en microcomputadoras, ha venido a llamarse RAM (memoria de acceso aleatorio). De aquí en adelante nos referiremos a RAM como una memoria programable.

Hay dos clases de memorias programables: estática y dinámica. La memoria programable estática almacena cada bit de información en un elemento de memoria biestable, tal como un circuito flip-flop. Esta información es retenida tanto tiempo como se mantenga la alimentación suministrada al circuito. Las memorias programables dinámicas tienen una estructura interna más simple, un tamaño más pequeño, disipan menos potencia y son intrínsecamente más rápidas. Almacenan información como una carga eléctrica en la puerta del sustrato del transistor MOS. Esta carga dura solamente unos pocos milisegundos y debe ser regenerada. Esta necesidad de regenerar la información almacenada es una de las más grandes diferencias entre las memorias programables estáticas y dinámicas.

El regenerar la memoria dinámica puede ser de algún modo molesto. El proceso requiere que todos los elementos de almacenaje estén direccionados por lo menos una vez cada pocos (por lo general 2) milisegundos. Un circuito contador suele incorporarse para ejercitar las líneas de dirección de la memoria, cuando la computadora no está teniendo acceso a la memoria. En la mayoría de los sistemas, la regeneración de la memoria requiere de una circuitería externa adicional. El Z80 contiene una circuitería del chip del procesador central y facilita, en gran manera, el uso de la memoria dinámica; no obstante, esta capacidad se pierde cuando se reinicializa el Z80. Por consiguiente, la circuitería de regeneración adicional es necesaria.

La elección entre la tecnología de memoria programable estática y dinámica, se basa en el coste y en la comodidad. Aun con el gasto de una circuitería externa de regeneración, la memoria dinámica es menos cara. En un sistema prototipo como el ZAP, no obstante, la memoria dinámica es más problemática que eficaz. Una vez construida y trabajando, la memoria dinámica podría ser perfectamente bien la mejor respuesta para la expansión de la memoria. Pero en este punto del proceso de construcción, la inclusión de la memoria dinámica podría complicar demasiado el diseño. Este libro, que es principalmente para principiantes, trata exclusivamente de aplicaciones de memoria programable estática de semiconductores.

Memoria programable estática

La figura 4.31 es un diagrama de bloques de un típico elemento de memoria programable estática del tipo empleado en la computadora ZAP. Hay cinco componentes básicos de una memoria programable: 1) línea de entrada de dirección, 2) entrada de datos, 3) salida de datos, 4) selección de chip y 5) una línea de strobe de habilitación de escritura o de lectura/escritura. Las líneas de entrada de dirección están conectadas al bus de direcciones de la computadora. En el caso de una memoria programable N por M bits, en donde N es el número de palabras y M es la longitud de cada palabra, deben haber suficientes líneas de dirección para direccionar todos los N octetos. Por ejemplo, en una memoria programable de 1 K tomaría 10 bits para direccionar todos los 1024 octetos dentro de esta memoria

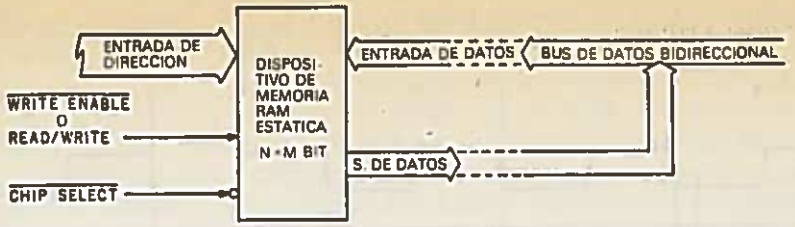


Figura 4.31 *Un diagrama de bloques de un elemento de memoria programable estática de N×M bits.*

(por ejemplo: $2^{10}=1024$). Los chips de memoria estática programable que contienen menos octetos de datos, tales como una memoria programable de 64 octetos, podrían requerir obviamente menos líneas de dirección. Para una memoria de 64 octetos, solamente son necesarios 6 bits de dirección.

Puesto que la función de un dispositivo de memoria programable estática es permitir almacenar y recuperar datos, debe proveerse la introducción y salida de datos desde el dispositivo. La línea de entrada y salida de datos (mostradas en la figura 4.31) están designadas como funciones separadas.

Durante la función de lectura, los datos almacenados dentro del elemento de memoria direccionado están disponibles en la línea de salida de datos. Durante la función de escritura, los datos que están ubicados en las líneas de entradas de datos podrían ser almacenados en la dirección indicada por el código en las líneas de entrada de dirección. No es necesario que los dispositivos de memoria estática programable tengan líneas independientes de entrada y salida de datos.

En la mayoría de los casos, estos dispositivos están configurados con salidas de «tres estados». Entrada y salida de datos pueden unirse a un bus de datos bidireccional, o pueden ser las mismas líneas y multiplicadas en el tiempo. La figura 4.31 ilustra un método de tres estados de configuración de buses de datos. Durante una función de lectura, las líneas de entradas de datos están incapacitadas dentro del dispositivo de memoria. Los contenidos del elemento de memoria provistas de dirección por las líneas de entrada de dirección están disponibles en la salida de datos y se alimentan directamente en el bus de datos bidireccional. Durante una función de escritura, lo contrario es cierto. Las líneas de salida de datos están puestas en el modo de tres estados (que es prácticamente un circuito abierto), y no extrae corriente del bus de datos bidireccional. El contenido del bus de datos bidireccional se almacena en el elemento de memoria indicado.

Todas las funciones multiplexadas son dependientes de las líneas de lectura/escritura y de selección de chips. Ninguna operación puede realizarse sin que el dispositivo de memoria se seleccione a través de la línea de «chip-select». Para seleccionar un banco particular, como se indicó antes, es necesario tener una lógica de decodificación que habilite estos bancos a través de las líneas de selección de chips. Una vez que se ha seleccionado un chip o un banco de chips, la computadora determina si los datos deben ser leídos de, o escritos en, estas posiciones de memoria. En condiciones normales, toda memoria programable estática se deja en el estado de lectura, y solamente se habilita durante la orden de escritura poniendo un nivel «0» en el terminal de habilitación de escritura («write-enable»). Esta es llamada señal strobe de habilitación de escritura.

La figura 4.32 es un cronograma detallado de los ciclos de lectura y escritura de la memoria. «Write/enable» (escritura/habilitación) es una combinación de petición de memoria y escritura. «Read/enable» (lectura/habilitación) es una combinación de petición de memoria y lectura. Ya se estudiaron con anterioridad la decodificación adecuada de estas señales y la selección de chips. En su forma básica, ZAP tiene 8 líneas de selección de chips, cada una de las cuales direccionan un banco de memoria de 1 K.

La figura 4.33 ilustra el mapa de memoria de la computadora básica ZAP. Según la configuración inicial, el ZAP contiene 3 koctetos de memoria. La posición

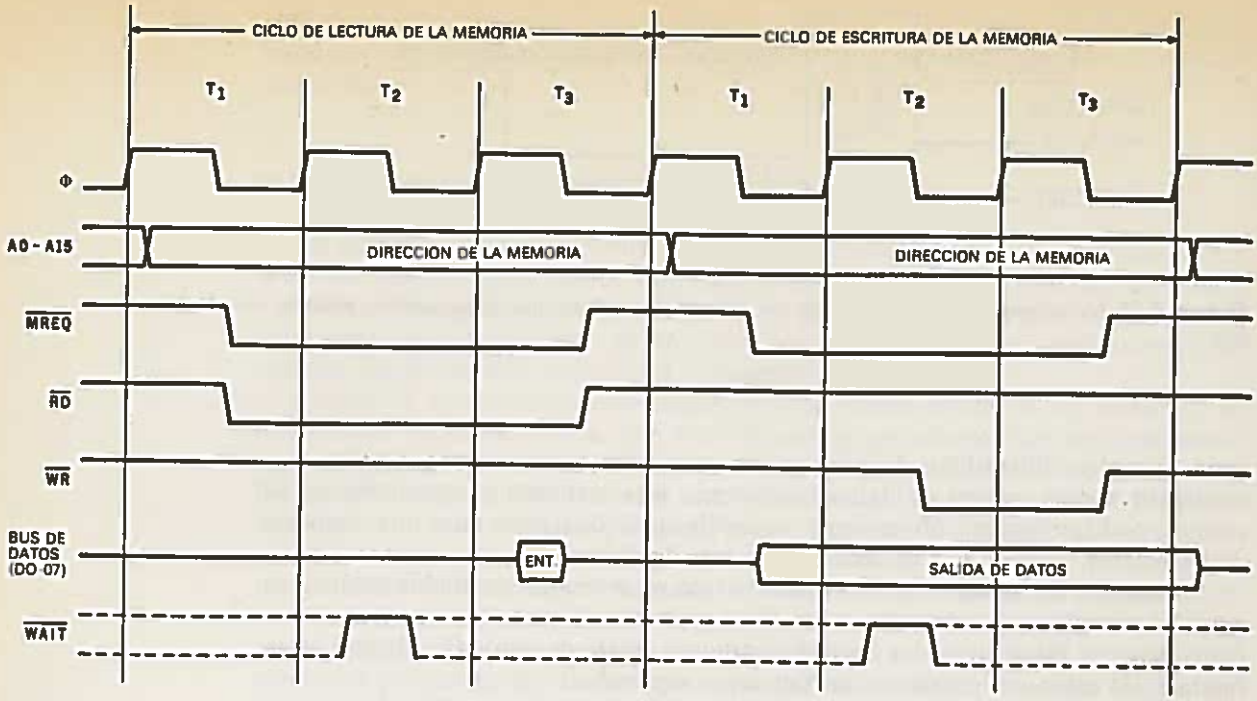


Figura 4.32 *Un cronograma de los ciclos de lectura o escritura de memoria para el Z80. Este diagrama no incluye los estados* $\overline{WAIT}$.

0 hasta 3FF es una EPROM de 1 K. Las posiciones 400 hasta BFF son posiciones de memoria programable estática. La EPROM de 1 K está diseñada para residir en posiciones de 0 hasta 3FF, de tal forma que el ZAP pueda ponerse en servicio fácilmente como una reinicialización de la alimentación. La memoria programable, ubicada en las posiciones de 400 en adelante, está considerada como memoria programable por el usuario. Por lo menos 2 K se recomiendan para trabajar satisfactoriamente. El ZAP trabajará con 1 K, pero 2 K está recomendado para una expansión de periférico básico.

La figura 4.33 muestra también cómo la memoria está unida a la computadora. Los tres bancos de memoria están unidos en paralelo entre los buses de datos y de direcciones. Cada banco tiene una decodificación individual para selección de chips («chip-select»). Cuando la EPROM está activada y $\overline{MCS0}$ está a un nivel lógico «0», los datos de EPROM son objeto de impresión en la línea de bus de datos. Los otros dos bancos de memoria están en el modo de tres estados y no tienen ningún efecto sobre el bus. Cuando la computadora tiene acceso a la memoria programable, el chip para ese banco particular de memoria se pone al nivel lógico «0» y solamente ese banco de memoria tiene acceso al bus de datos.

Aunque todos los bancos de memoria tuvieran la misma dirección aplicada a los mismos, sólo el banco seleccionado estaría en el modo activo. El flujo lógico es similar, para la computadora, a escribir en un banco de memoria. Observará que hay línea de habilitación de escritura que llevan a cada uno de los bancos de memoria programable estática de 1 K, pero no a la EPROM de 1 K. Esta última sólo puede escribirse con un interface especial. Por consiguiente, sólo las memorias programables están unidas a strobe de habilitación de escritura.

Si, por ejemplo, la computadora tuviera que escribir en la posición 400, la selección de chips para el banco 1 y la habilitación de escritura para el banco 1 estarían ambas a nivel lógico «0» para permitir que los datos en el bus de datos se

almacenen en la posición 400. Este tipo de configuración de memoria programable está multiplexada y en el modo de tres estados. En el modo de lectura, los datos fluyen desde el chip de memoria programable; en el modo de escritura fluyen hacia dicho chip, y cuando no está seleccionado está en el modo de tres estados.

Hasta este punto, hemos examinado los diagramas de bloques de memoria programable estática. Para obtener una computadora operacional, es necesario configurar esta memoria con partes reales. Lamentablemente, las memorias programables de 1 K×8 bits de una sola pastilla integrada eran muy caras cuando se diseñó el sistema ZAP. Por consiguiente, estos bloques de 1 K están diseñados a partir de componentes múltiples. Dos chips de memoria programable estática populares y baratos son el 2102A (Apéndice C3) y 2114 (Apéndice C4) de Intel.

2102A es una memoria programable estática de 1 K×1. La configuración de un bloque de memoria de 1 K×8 requiere ocho circuitos integrados 2102 unidos en paralelo. Para comparación, una configuración para un bloque de 1 K×8 usando 2114 requeriría solamente de dos chips. Esto es porque el 2114 tiene una densidad interna más alta que la del 2102. Como el objetivo de cualquier proyecto de computadora cableada a mano es conseguir fácilmente el dispositivo en línea, los 2114 son los dispositivos de memoria programable recomendados para ZAP. Aunque los 2102 trabajaran adecuadamente, el cableado adicional necesario para utilizarlos compensará con mucho el coste adicional de los 2114.

La figura 4.34 ilustra cómo dos 2114 están conectados juntos para producir un banco de memoria programable de 1 K×8. Ellos comparten una línea común de selección de chips. Las líneas de entradas de datos están divididas de tal forma que 4 bits de datos estén almacenados en cada chip. Como cada una tiene una capacidad de dirección de 1024 octetos, las líneas de dirección de 10 bits suelen estar compartidas. Para construir el ZAP básico, dos circuitos como los que se muestran en la figura 4.34 deben ser construidos. La memoria total para la computadora básica es de 3 K. Puede ser ampliada a 8 K sin decodificación adicional de dirección. No es absolutamente necesario poseer una memoria programable de 2 K si el usuario sólo desea revisar el funcionamiento del sistema. En un mínimo, la EPROM debe ser cableada como un banco de memoria.

La EPROM de 1 K contiene el monitor que permite al ZAP funcionar. Este monitor contiene muchos programas más pequeños que son llamados subrutinas. Cuando un programa principal llama a una subrutina, coloca la dirección de retorno en una pila software ubicada en la memoria programable. A la conclusión de la subrutina, el procesador central saca esta dirección de la pila y vuelve al programa principal. Por lo general, la pila no requiere más de 64 octetos. Sin embargo, no es menos complicado cablear dos 2114 para un banco completo de memoria de 1 K×8 que tratar de cablear una memoria de 64 octetos.

Un banco adicional de 1 K, designado como banco dos, podría añadirse a discreción del usuario. Este banco es necesario si se planea escribir programas que ocuparán más de 1 K de memoria la pila. Como se concibe la computadora, en este libro, 1 K puede parecer adecuado. Sin embargo, los programas adicionales bosquejados en el libro sería recomendable utilizarlos con 12 K. Esto es cierto sobre todo cuando un área de buffer se precisa para comunicar con periféricos externos. El esquema para la configuración final de la memoria se muestra en la figura 4.35. Debe añadirse a la circuitería de las figuras 4.17 y 4.26.

A diferencia con las otras secciones de la computadora, la memoria no puede ser revisada excepto bajo control del programa. Teóricamente, las líneas de dirección pueden prefijarse y los datos leerse o almacenarse, pero no vale la pena el esfuerzo. Las comprobaciones de memoria se producirán después de que se cablee la sección de entrada/salida. Básicamente, se comprobará primero con EPROM sola y luego con la adición de la memoria programable. Se indicó anteriormente que la EPROM y la memoria programable están relacionadas aunque funcionan con independencia. Aunque un programa se suele almacenar en PROM, habitualmente requiere memoria programable para una ejecución adecuada.

En un programa corto que carga el acumulador, escribe a un port de salida y vuelve a sí mismo de nuevo, sin llamadas de subrutina, la memoria programable no es necesaria. Puede estar completamente ubicada en EPROM. El procedimiento exacto para esta prueba será bosquejado al final de la sección E/S.

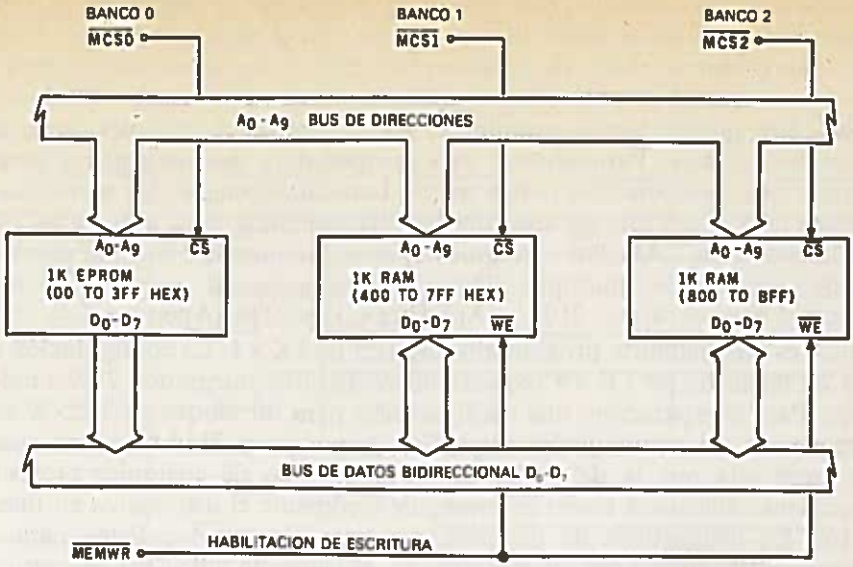


Figura 4.33 *Un diagrama de bloques del mapa de la memoria de la computadora ZAP.*

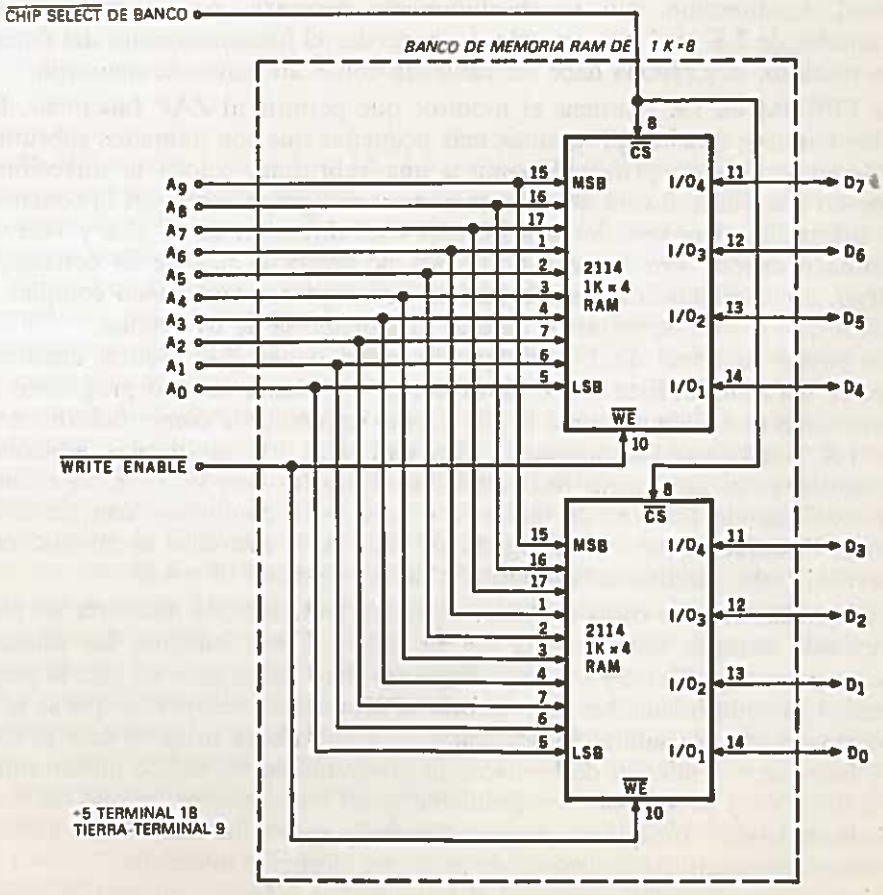


Figura 4.34 *Un banco de memoria programable de 1 K×8 construido utilizando dos chips 2114 de memoria programables de 1 K×4 bits.*

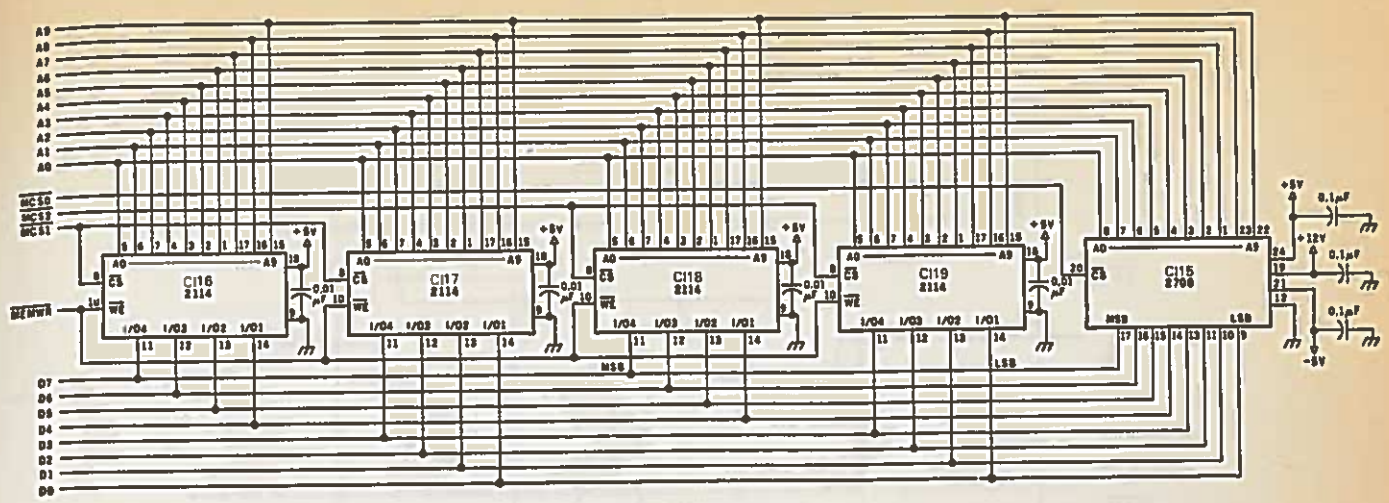


Figura 4.35 *Un diagrama esquemático de la configuración final de la memoria para la computadora básica ZAP.*

IV. Entrada/Salida

Hasta ahora, hemos examinado el control del procesador central y la decodificación de la memoria. Las funciones de entrada y de salida son igualmente importantes. Para que la computadora presente visualmente información de utilidad, debe estar interconectada con periféricos. «Interfaz» es un término que se usa en demasía y que indica una capacidad de comunicación con dispositivos externos, tales como teclados, vídeo o visualizadores de LEDs y sistemas de almacenamiento de memoria. La comunicación puede ser o la entrada o la salida de datos.

Los datos de entrada pueden proceder de teclados, de almacenamiento de casete de audio o de interfases especiales de adquisición de datos. De manera similar, los datos de salida fluyen desde la computadora a periféricos (por ejemplo: pantallas de vídeo, salida de lecturas numéricas, impresoras e interfases externos de control). La función y el formato de la comunicación de datos entre el procesador central y los periféricos podría variar considerablemente, pero el encaminamiento interno de los datos fundamentalmente el mismo.

El microprocesador Z80 proporciona instrucciones de entrada y salida. Una salida del procesador es desde el punto de vista lógico igual como escribir a la memoria, y escribir una entrada de un dispositivo externo es similar a una orden de lectura de memoria. Están diferenciadas de las operaciones de memoria seleccionando la señal de las líneas de lectura y escritura con la línea de control de petición de E/S. La coincidencia lógica de una petición de E/S y una condición de salida lectura o escritura indica el sentido de la comunicación con el dispositivo periférico. Simultáneamente con las señales de control, el código de dirección (1 de 256) del dispositivo en cuestión es colocado en el bus de direcciones. Un cronograma de estas señales es mostrado en la figura 4.36. La lógica de decodificación fue detallada en la sección II de este capítulo.

El cableado de los ports de E/S para ZAP es un proceso de dos etapas. Cuando se cablea a mano la computadora, la consideración más importante es ver que la función de entrada/salida trabaje de la forma menos complicada. Una prueba satisfactoria de la sección de E/S del ZAP prueba también indirectamente la memoria. Esto es así porque las instrucciones de entrada y de salida no pueden ser ejercitadas excepto por un programa almacenado en la memoria.

La entrada y la salida del Z80 tratan 8 bits a la vez. No importa si el interfaz externo está en serie o en paralelo. La transferencia de datos entre el procesador central y el dispositivo de E/S es de 8 bits en paralelo y básicamente se produce como sigue:

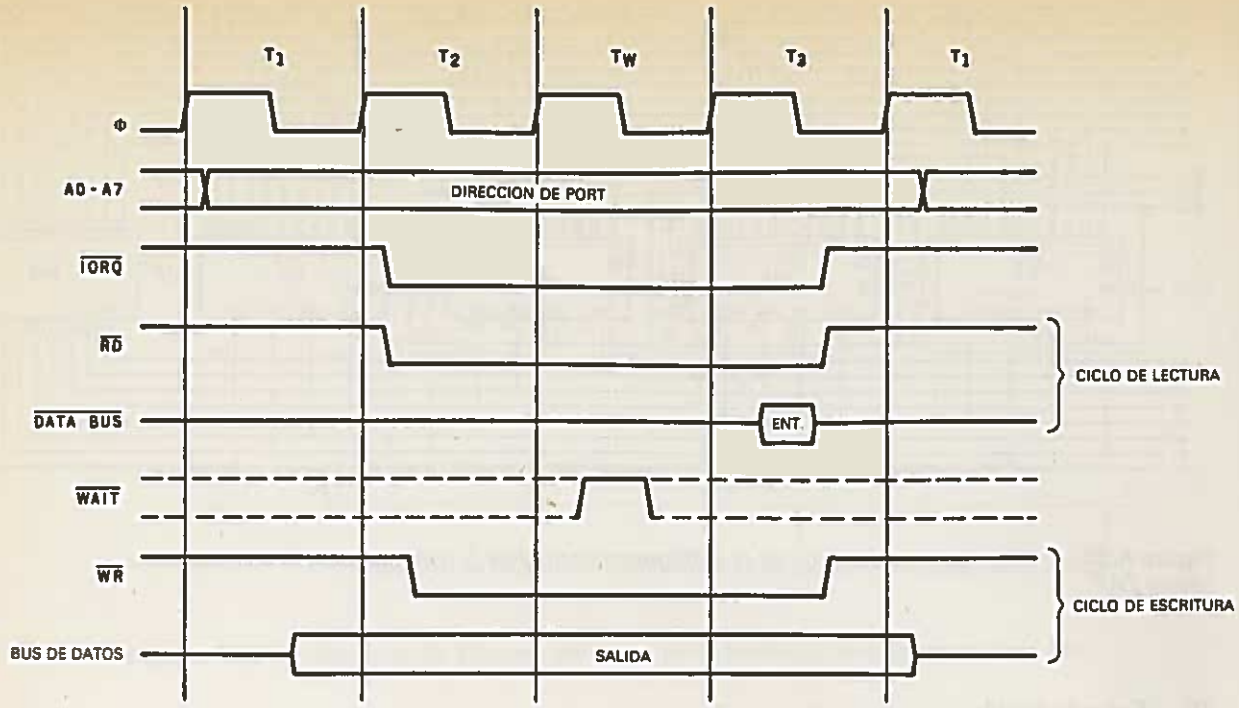


Figura 4.36 *Un cronograma de los ciclos de entrada o de salida para el Z80.*

Instrucciones de salida

OUT(n), A

Cuando se ejecuta esta instrucción, el contenido del acumulador se coloca en el bus de datos y se escribe en el dispositivo n. La dirección del dispositivo está localizada en las líneas de dirección A0 hasta A7.

Si el acumulador contiene 40 en código hexadecimal cuando se ejecuta la instrucción OUT 23, A, 40 hexadecimal será escrita en el dispositivo periférico (también llamado «número del port») decodificado como 23 hexadecimal.

Aunque hayan instrucciones de salida más complicadas en el juego de instrucciones del Z80, todas pasarán datos, a través del bus de datos, hacia el dispositivo externo. Puesto que el bus de datos es utilizado para transferencia de información entre el procesador central y la memoria. Así como con E/S, la computadora debe estar en libertad de continuar la ejecución de su programa. Los datos no pueden permanecer en el bus de datos esperando por el periférico (el procesador central puede estar hecho para hacer esto, pero tales configuraciones abstractas podrían confundir en esta ocasión). Los datos son válidos solamente para unos pocos ciclos de reloj y deben ser almacenados, si es necesario, durante un período más largo.

La figura 4.37 muestra un típico registro de almacenamiento de 8 bits. Consiste en 8 elementos de almacenaje individuales con una entrada común de «store enable» (habilitación de almacenamiento). En su forma más simple, los elementos sencillos de almacenaje pueden ser flip-flops tipo D, tales como los indicados en la figura 4.38. Los datos de entrada (es decir, el bus de datos) están unidos a las líneas de entrada D y sólo se sincronizan con la señal de reloj en las líneas de salida (Q y $\overline{Q}$) durante una señal strobe de escritura de E/S. Con el empleo de los circuitos 7474 se precisarían 4 chips para una palabra de 8 bits. Un método más conveniente es utilizar los circuitos perfeccionados de la figura 4.39.

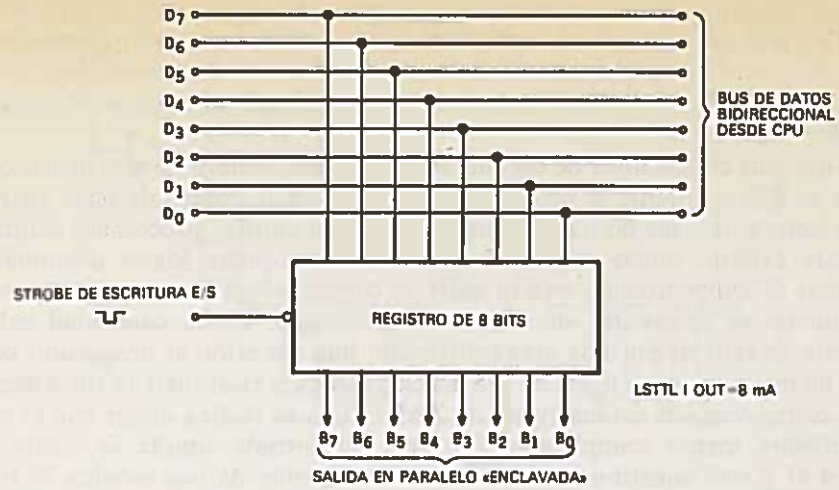


Figura 4.37 *Un diagrama de bloques de un típico port de salida en paralelo «enclavado» configurado con un registro de almacenamiento de 8 bits.*

Instrucciones de entrada

IN A, (n)

Cuando se ejecuta esta instrucción, los datos procedentes del port seleccionado (n) se colocan en el bus de datos y se cargan en el acumulador.

Si dicho dispositivo externo lee 10 hexadecimal cuando la instrucción IN A, se ejecuta 20 y el valor 10 hexadecimal leído a partir del número de dispositivo 20 hexadecimal sería cargado en el acumulador.

Hay otras instrucciones de entradas más complicadas, pero como en el caso de las instrucciones de salida, la ruta para todos los datos sigue siendo el bus de datos. Para mantener el bus de datos controlado por un dispositivo simple conectado al mismo, todos los dispositivos de entrada (es decir, la salida de los mismos) deben

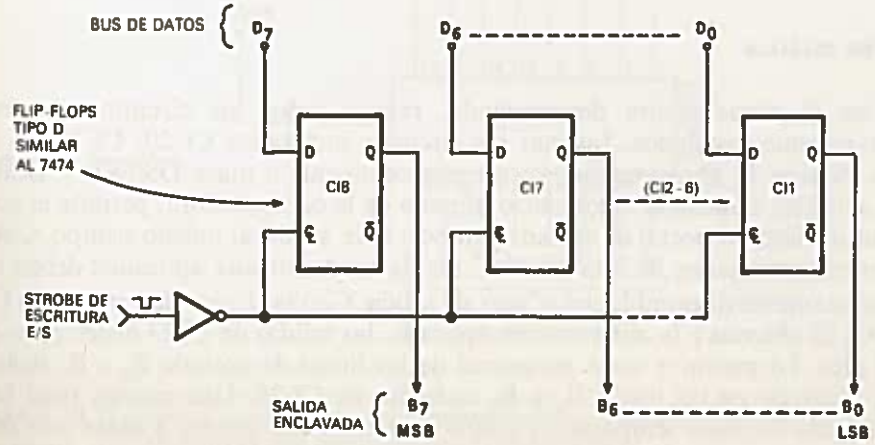


Figura 4.38 *Diagrama de bloques de un port de salida en paralelo enclavado utilizando flip-flops tipo D como un registro de almacenamiento.*

ser de tres estados. Esto puede realizarse utilizando lógica de interfaz, tal como UARTs y adaptadores de interfases de periféricos (PIAs) que están diseñados para ser de «tres estados» o bien añadiendo buffers de entrada de tres estados, tal como se ilustra en la figura 4.40 (el diagrama de bloques del port típico de entrada en paralelo y de 8 bits).

Lo que esté en las líneas de entrada B_0 a B_7 inclusive durante una instrucción de lectura de E/S se dirigirá al procesador central. Con el empleo de estas instrucciones de lectura directas no hay ninguna interacción entre el procesador central y el hardware externo unido al port de entrada. Se requiere lógica adicional para coordinar la temporización exacta entre la computadora y un periférico externo. La solución se denomina «handshaking» (diálogo). Dicha capacidad exige un hardware de port de entrada más sofisticado, una conexión al procesador central, lógica de interrupción o ports de E/S adicionales para coordinar la sincronización.

La comprobación del hardware de ZAP básico se realiza mejor con el empleo de hardware menos complicado. Un port de entrada simple se ilustra en la figura 4.41 y está constituido por 2 buffers cuádruples de tres estados. Si hubiera algún experimentador audaz que deseara establecer un diálogo completo en ports de E/S o necesitara capacidades de control de salida de más de 8 mA de un dispositivo LSTTL, los ports de entrada y de salida pueden configurarse fácilmente con el empleo del 8212 de Intel. Las especificaciones descritas en el Apéndice C5 ponen de manifiesto su versatilidad.

Comprobación de entrada/salida

Ultimamente, ZAP podría tener un teclado, un terminal de TRC en serie RS232, un interfaz de casete de audio y capacidades de E/S analógicas así como digitales. El intento de unir todos estos periféricos juntos y de comprobar todo simultáneamente sería complicado. Un enfoque más metódico es construir los hardware y software mínimos que demuestren ser suficientes para el funcionamiento adecuado y luego seguir la construcción sobre esa base. Ese es el camino tomado hasta ahora.

Con la excepción de la memoria, hemos intentado eliminar cualquier posible problema mediante pruebas estáticas, siempre que sea viable. Los dispositivos de E/S simples de las figuras 4.39 y 4.41 se prestan fácilmente a esta situación. Para una prueba completa de E/S se requiere un port de entrada y un port de salida. Debe cablearse como se indica en la figura 4.42. Solamente el port 0 necesita conectarse en este momento. Puede ignorarse la circuitería adicional incluida en este diagrama. Solamente los circuitos integrados 21 a 23 inclusive son de interés actualmente. Los demás dispositivos son mejoras del ZAP básico y se examinará más adelante.

Prueba estática

Con la alimentación desconectada, retirar todos los circuitos integrados anteriormente instalados. Insertar los circuitos integrados CI 20, CI 21, CI 22 y CI23. Aplicar la alimentación. Poner temporalmente a masa $\overline{\text{DS0WR}}$ y $\overline{\text{DS0RD}}$. Este artificio, imposible bajo control directo de la computadora, permite el acceso de bus de datos al port 0 de entrada y al port 0 de salida al mismo tiempo. Con los dos ports conectados de esta manera, los datos de entrada aplicados deben estar inmediatamente disponibles en el port de salida. Con las líneas de entrada de CI 21 y de CI 22 abiertas y la alimentación aplicada, las salidas de CI 23 deben estar a un nivel alto. La puesta a masa secuencial de las líneas de entrada B_0 a B_7 inclusive debe reflejarse en las líneas B_0 a B_7 inclusive de CI 23. Una prueba final ha de desconectar la masa temporal en $\overline{\text{DS0WR}}$ aunque esté puesto a masa una de las líneas de entrada de CI 21 y de CI 22. La salida de nivel lógico «0» de CI 23 debe mantenerse a nivel bajo incluso cuando la línea de entrada ya no está puesta a tierra. El resultado es que los datos estén «enclavados». Se mantendrán así hasta que se actualice por otra señal strobe de escritura.

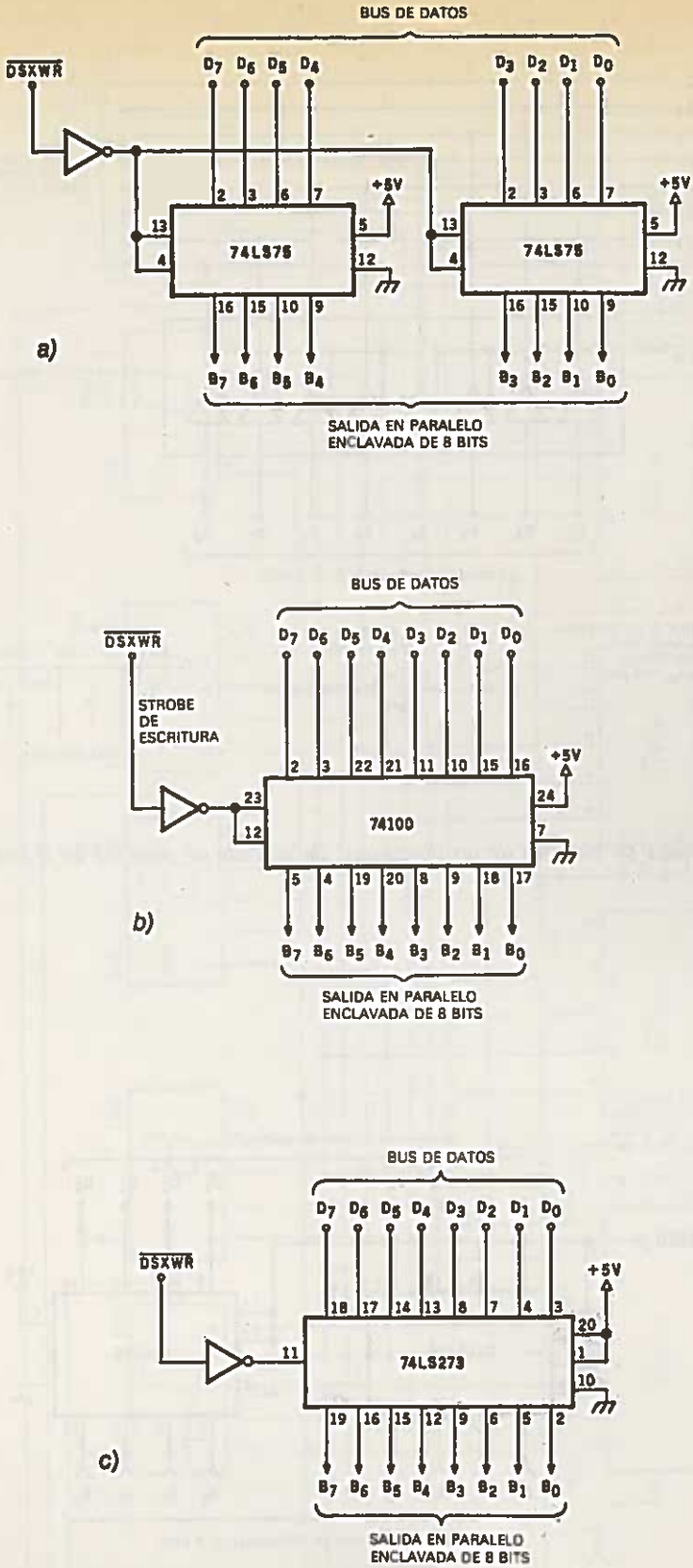


Figura 4.39 *Diagramas esquemáticos de ports de salida en paralelo enclavados de 8 bits.*

a) Utilizando dos enclavadores de 4 bits LSTTL.

b) Utilizando un enclavador tradicional de 8 bits TTL. Observe que los dispositivos no LSTTL pueden ser sustituidos, pero debe tenerse cuidado en observar la carga total del bus.

c) Utilizando un nuevo enclavador de 8 bits LSTTL.

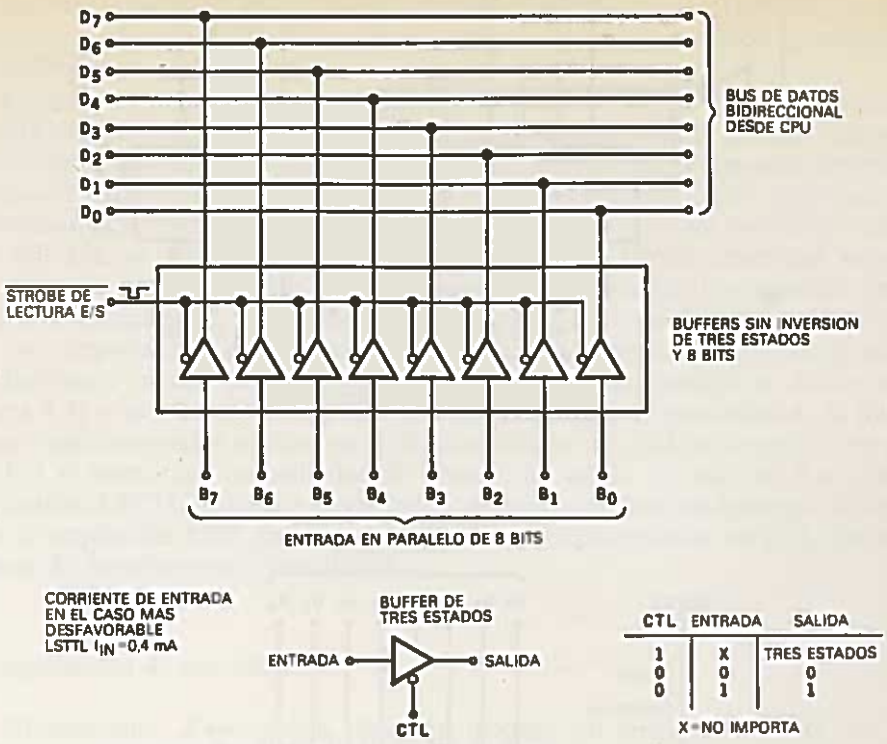


Figura 4.40 *Un diagrama de bloques de un típico port de entrada en paralelo de 8 bits.*

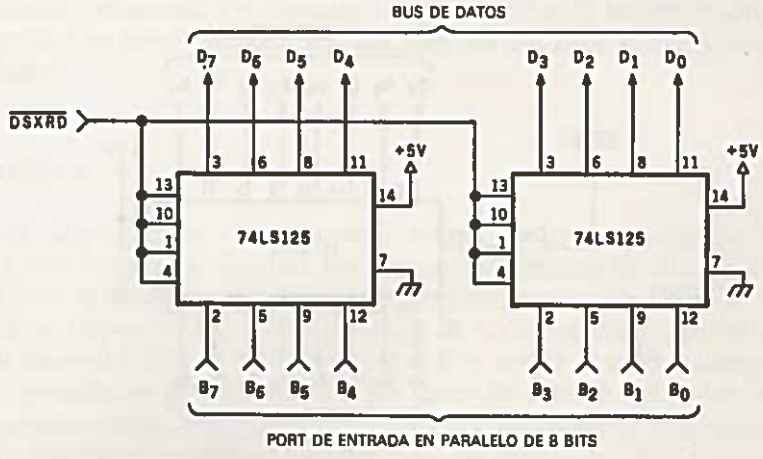


Figura 4.41 *Un diagrama esquemático de un port de entrada en paralelo de 8 bits para la computadora ZAP.*

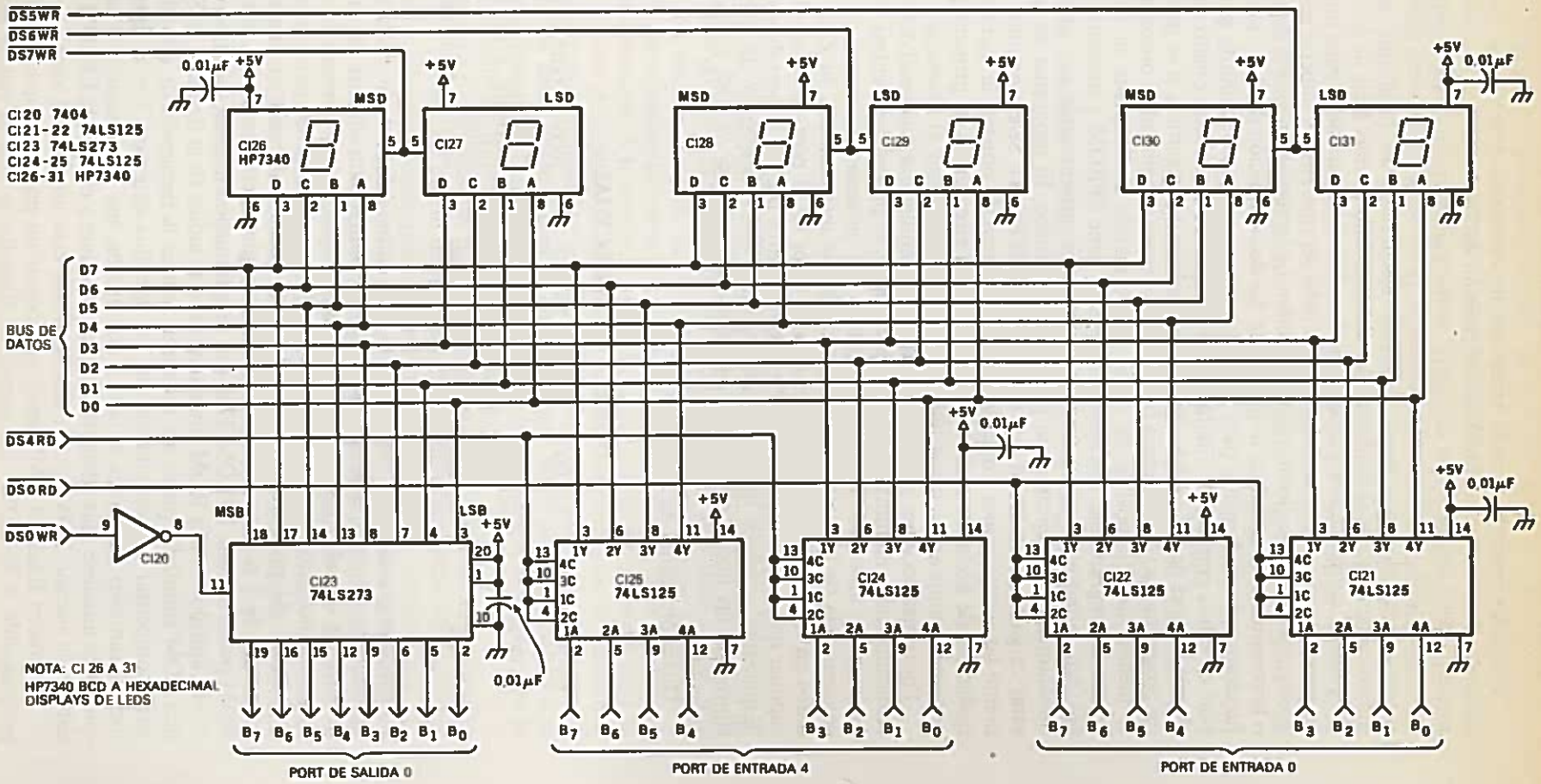


Figura 4.42 *Un diagrama esquemático de ports de entrada/salida en paralelo de la computadora ZAP básica con mejoras adicionales necesarias para utilizarlas con el software de monitor de ZAP.*

V. Comprobación dinámica de la computadora básica

Todos los sistemas, con la excepción de memoria, deben haber pasado satisfactoriamente los procedimientos de revisión de tipo estático. El cableado de la memoria debe comprobarse en lo que respecta a su continuidad. Puesto que ZAP no tiene ningún panel frontal ni indicador (a no ser que se quiera añadir uno), el sistema completo sólo puede probarse ejecutando un programa que ejercite dinámicamente todo el hardware del sistema. Ello es más fácil de lo que parece. Para que la computadora proporcione la salida de un número para una dirección de port específica, el procesador central debe ser operativo y haberse reinicializado adecuadamente para ejecutar la instrucción. La lectura de memoria debe funcionar o la computadora no sabría lo que hacer. La decodificación de E/S y memoria debe funcionar adecuadamente para los datos almacenados en memoria para llegar al port de salida correcto. Y, finalmente, para que los datos sean objeto de lectura en el port, el port de salida debe funcionar también. En resumen, si se puede ejecutar un programa, la computadora funciona. Podemos simplificar el proceso utilizando el menor número posible de pasos y eliminando inicialmente la necesidad de memoria programable. Recuérdese que ZAP tiene EPROM y memoria programable. Sin monitor ni panel frontal, la memoria programable no puede cargarse directamente para ejecutar un programa de prueba. El programa de prueba debe estar cargado ya en ROM (en nuestro caso, EPROM). Seleccionando cuidadosamente las instrucciones utilizadas en el programa de prueba, la memoria programable puede excluirse completamente cuando ejecutamos la primera prueba. ¿Y por qué complicar las cosas teniendo más hardware que el necesario?

Pocas instrucciones se requieren para la computadora funcional del procesador, reinicialización, memoria y E/S. Normalmente, el procesador central trabaja o no lo hace en absoluto. El fallo operativo del procesador central es, rara vez, consecuencia de la ejecución inadecuada de las instrucciones. Si ZAP puede leer datos en el port 0 y hace salir el mismo valor en el port de salida 0, podemos suponer que todo está en orden. Para que los datos lleguen al port de salida 0, debe desplazarse a través del procesador central (en el supuesto de que hayamos suprimido las masas temporales en las líneas de strobe de E/S) bajo control del programa.

Dicho programa de prueba es:

		OCTAL	HEXADECIMAL
IN A, 0	333 000	DB 00	lectura port 0 de entrada
OUT 0, A	323 000	D3 00	escritura a port 0 de salida
JP NN	303 000 000	C3 00 00	salto al principio

Este programa de 7 octetos leerá los datos del port de entrada 0 en el acumulador y luego escribirá estos mismos datos para el port de salida 0. La instrucción de salto hará que el programa repita continuamente la acción. El programa no requiere memoria programable para almacenar datos intermedios o el puntero de pila. Puesto que solamente resulta afectado el acumulador, el programa de 7 octetos puede contenerse completamente en ROM. En este caso, ROM puede ser una EPROM 2708 con programación manual tal como se describe en el Capítulo 7 o una ROM simulada como se indica en la figura 4.30. Si se utiliza una ROM simulada, puede ser necesario reducir la frecuencia de reloj de 2,5 MHz para compensar la capacitancia de los circuitos externos. En la figura 4.30 se incluye también una salida al port 5 que prueba una presentación visual de datos que ha de añadirse más adelante. Mejor que volver a escribir la EPROM o volver a cablear la pseudo ROM, puede desear añadir esta instrucción ahora.

La prueba final del ZAP básico es ejecutar un programa que utilice memoria programable y EPROM. De nuevo, la filosofía es que si puede almacenar y recuperar 1 octeto de la memoria programable, entonces toda la capacidad de 1 K debe actuar. En esta ocasión, se emplea un programa algo más largo. El siguiente

programa se almacena en EPROM y la memoria programable se emplea por el procesador central para almacenar la pila:

		OCTAL	HEXADECIMAL	
	LD SP, nn	061 000 006	31 00 06	poner puntero de pila a mitad de memoria programable de banco 1
	IN A, 0	333 000	DB 00	lectura de port 0, de entrada
	CALL TEST	315 014 000	CD 0D 00	llamada prueba de programa
	OUT 0, A	323 000	D3 00	escritura datos a port 0, salida
	JP nn	303 000 000	C3 00 00	salto al principio
TEST	RET	311	C9	retorno a programa principal

Cuando se ensamble el programa de 14 octetos se cargaría como sigue (en hexadecimal):

Posición	Programa
00/00	31 00 06
03	DB 00
05	CD 0D 00
08	D3 00
0A	C3 00 00
0D	C9

El funcionamiento de este programa es similar al ejemplo anterior. Un octeto es objeto de lectura a partir del port de entrada 0 y luego retorna la lectura al port de salida 0. Entre estas operaciones hay una llamada a una subrutina que es una instrucción de retorno. Cuando se ejecuta la llamada, la posición en donde el programa ha de reanudar la operación, después de la llamada, se lleva a la pila en memoria programable. A la conclusión de la llamada (la instrucción de retorno), la dirección se extrae de la pila y se coloca en el contador de programa, de modo que el programa pueda reanudarse en donde se dejó. La única forma para la introducción de datos desde el port de entrada 0 para llegar al port de salida 0 es mediante esta llamada a ejecutar adecuadamente. Por supuesto, ello exige que la memoria programable trabaje de forma apropiada.

Muchos otros programas que mejorarían los procedimientos de comprobación de diagnóstico pueden ser objeto de escritura. Según mi experiencia, sin embargo, si ejecuta estos dos programas, puede contar con que todo está funcionando.

Una vez llegado a este punto, el experimentador tiene una computadora verdaderamente operativa. El siguiente paso es ampliar esta unidad básica y hacer del ZAP un sistema algo más versátil añadiendo visualizaciones de datos y direcciones, un teclado hexadecimal, un interfaz en serie, junto con un sistema operativo que coordine las actividades de estos periféricos. Aunque el sistema actual es una computadora, estas adiciones son necesarias para ir más allá de un simple proyecto meramente experimental.

Capítulo 5
LOS PERIFERICOS BASICOS

Una vez que ha sido construida y probada la computadora básica ZAP, estamos en condiciones para agregar unos pocos periféricos necesarios que aumentarán la utilidad del sistema. Los periféricos externos facilitan las capacidades de entrada y salida de la computadora. Incluyen elementos tales como impresoras, tubos de rayos catódicos (TRC), unidades de disco y discos. Periféricos de esta magnitud suelen emplearse, sin embargo, en sistemas más grandes. Para nuestra computadora ZAP, basada en el Z80, los periféricos útiles incluyen un teclado que facilita la introducción de datos y de programas; una pantalla de visualización para permitir a la computadora indicar una conclusión lógica de manera legible; una interfaz de comunicación en serie, que permite a la ZAP «hablar» a otra computadora, y una interfaz a un dispositivo de almacenaje de casete de audio. Estos cuatro son la diferencia entre un modelo experimental y una computadora personal útil.

El teclado puede ser o un pequeño teclado numérico, para una introducción de datos limitada, o un teclado tipo «máquina de escribir» alfanumérico ASCII (Código Americano de normas para intercambio de información), para la edición del texto y programación en lenguaje de alto nivel. La pantalla de visualización podría variar de una salida de lectura hexadecimal con diodos LED a un terminal de TRC de 24 líneas × 80 caracteres. El port en serie, en conjunción con el interfaz de audio en casete, podría ser usado para «arrancar en frío» la computadora y cargar programas de aplicación.

Como en los circuitos anteriores en este libro, se ha tratado de proporcionar varias alternativas de diseño de tal forma que se pueda construir un verdadero sistema personal. Cada uno de los cuatro dispositivos periféricos será explicado con detalle y serán proporcionados numerosos ejemplos de diseño; los teclados ASCII completos y de entrada hexadecimal de función limitada estarán direccionados. En el caso de la pantalla de visualización, estudiaremos una salida en octal elemental con LEDs y una salida de lectura hexadecimal para el ZAP. Para una visualización más sofisticada, se requiere un terminal TRC. Como esta unidad es mucho más complicada que un teclado o que una visualización con LEDs, se le ha dedicado un capítulo completo. Mi premisa básica es comenzar con lo esencial, hacer comprender a fondo sus aplicaciones y luego pasar a elementos adicionales complejos y de mayor utilidad.

La ampliación del ZAP básico es un sistema microcomputarizado interactivo, requiere la inclusión de un programa de software para sincronizar y hacer funcionar los nuevos periféricos. Este software se denomina monitor y es explicado en un capítulo posterior. Los periféricos sólo proporcionan los medios para añadir capacidad de visualización y de introducción de datos.

I. TECLADOS

La única forma en la que el Z80 se puede comunicar con dispositivos externos es a través de la estructura de buses de entrada/salida previamente descrita.

(Aunque existan más métodos esotéricos, tales como un acceso directo a memoria, serán ignorados de momento.) Cuando el procesador desea indicar al usuario que ha ocurrido un evento, puede hacerlo cambiando el nivel de salida de un bit de un port de salida en paralelo. Por ejemplo, la finalización de la ejecución de un programa puede ser indicada por el bit 7 en el port 0 pasando de un nivel lógico «0» a un nivel lógico 1. Utilizando este concepto, 8 elementos separados podrían estar individualmente indicados y controlados por los 8 bits de salida proporcionada en el port único del ZAP básico.

La entrada de la información es simple. Los números 0 hasta el 7 inclusive podrían corresponder a 8 interruptores en los 8 bits de entrada del port 0. Esto se ilustra gráficamente en la figura 5.1. Cuando se pulsa el interruptor del bit 7, poniendo a masa la entrada, la transición del nivel lógico puede indicar una entrada numérica de 7 a la computadora; muchas aplicaciones de microprocesadores requieren solamente estos pocos bits de E/S. Un controlador de luces de tráfico, por ejemplo, con luces roja, amarilla y verde, necesitaría sólo 3 bits de salida.

El programa para controlar las luces podría estar escrito, ensamblado y programado en algún tipo de almacenamiento no volátil. Sin embargo, el ZAP debe estar en interacción con el operador humano, de tal forma que los programas puedan ser desarrollados y probados. La mayor diferencia entre el controlador de tráfico y el ZAP serían los periféricos y no las capacidades del microprocesador.

En nuestro ejemplo, podríamos poner 8 interruptores en el port de entrada. Para introducir información sólo tenemos que escribir un pequeño programa que lea los datos en el port 0 dentro del acumulador y que luego los almacene o actúe sobre ellos. El capítulo acerca del software de monitor direccionará estas manipulaciones, pero un problema debe resolverse primero: la sincronización de los periféricos con la computadora.

¿Cómo conoce la computadora cuándo los datos en el interruptor son o no válidos? Y, ¿podríamos hacer un temporizador en software o hardware que lea el port cada segundo? ¿Puede, por ejemplo, imaginarse a sí mismo tratando de bascular todos los interruptores a la vez o hacer esperar a la computadora?

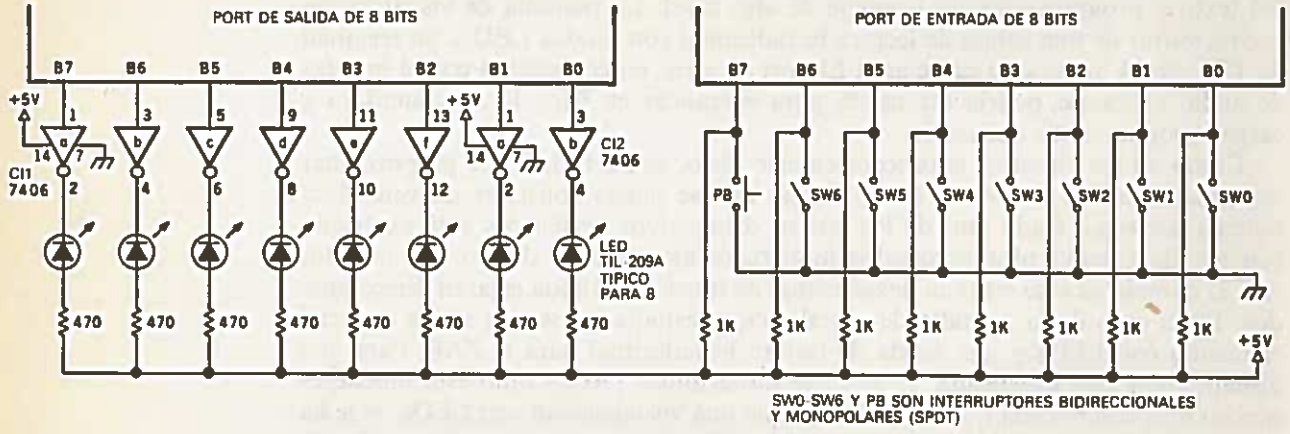


Figura 5.1 *Un interfaz de entrada/salida en paralelo con salida de lectura con LEDs y entrada por interruptores.*

El método más popular de sincronizar un periférico, que tiene una introducción lenta de datos a una computadora de ejecución rápida de programa, es utilizar pulsos de strobe (habilitación) de «datos listos». (Pueden utilizarse también interrupciones, pero implicarían una complicada programación y no serán considerados aquí.) El programa es escrito para leer y revisar el nivel lógico de un solo bit. Sustituyendo un pulsador por uno de los 8 interruptores, por ejemplo el bit 7, podemos simular el strobe. Para lograr esto, primero hay que poner a «1» los datos en los otros 7 interruptores; luego, con el programa puesto en un bucle de revisión del bit 7, oprimir el pulsador para generar una transición lógica. El programa, al

percibir que el strobe de «datos listos» está presente, lee en el port completo y utiliza los otros 7 bits de datos.

No suele ser práctico limitarnos nosotros mismos a sólo siete interpretaciones simbólicas cuando utilizamos 7 bits de entrada. Un método más lógico es codificar la entrada y permitir a los 7 bits representar a los 128 símbolos individuales. La elección entre una entrada codificada y una entrada en paralelo está basada principalmente por su aplicación. Si la computadora es parte de una alarma antirrobo, con cada bit de entrada representando un interruptor de puerta o ventana. Entonces es importante conocer transiciones de bits individuales y simultáneas. En esta aplicación es necesario tener una entrada de señal en paralelo. Por otra parte, la entrada alfanumérica por teclado tipo «máquina de escribir» está por naturaleza en serie con una letra cada vez. Por consiguiente, nada se gana con el empleo de 128 bits de entrada en paralelo para un teclado de 128 teclas. Un código de 7 bits tiene una mayor relación coste/prestación.

El código de teclado más utilizado es el ASCII (código americano de normas para intercambio de información). En el Apéndice B se indica el código y los caracteres que lo representan. Cualquier teclado casero debe utilizar este código para ser compatible con el software existente en el mercado, tal como BASIC.

Hay varios métodos que pueden ser utilizados para generar códigos de teclas apropiados. Las figuras 5.2 y 5.3 muestran métodos de hardware y de software, respectivamente. El diagrama de bloques bosquejado en la figura 5.2 es un sistema de exploración de hardware apropiado para un teclado de 64 teclas. Un contador de 6 bits habilita sucesivamente cada columna, mientras que explora todas las filas en cada paso. Si se pulsa cualquier tecla, un nivel lógico «0» será dirigido, a través del multiplexor de 8 entradas, a la lógica de control de exploración. Esta señal se utiliza para generar una «strobe» de tecla pulsada (también llamada strobe de «datos listos») para la computadora. Las líneas de dirección de las filas y columnas procedentes del contador son leídas e indican la dirección de la matriz binaria

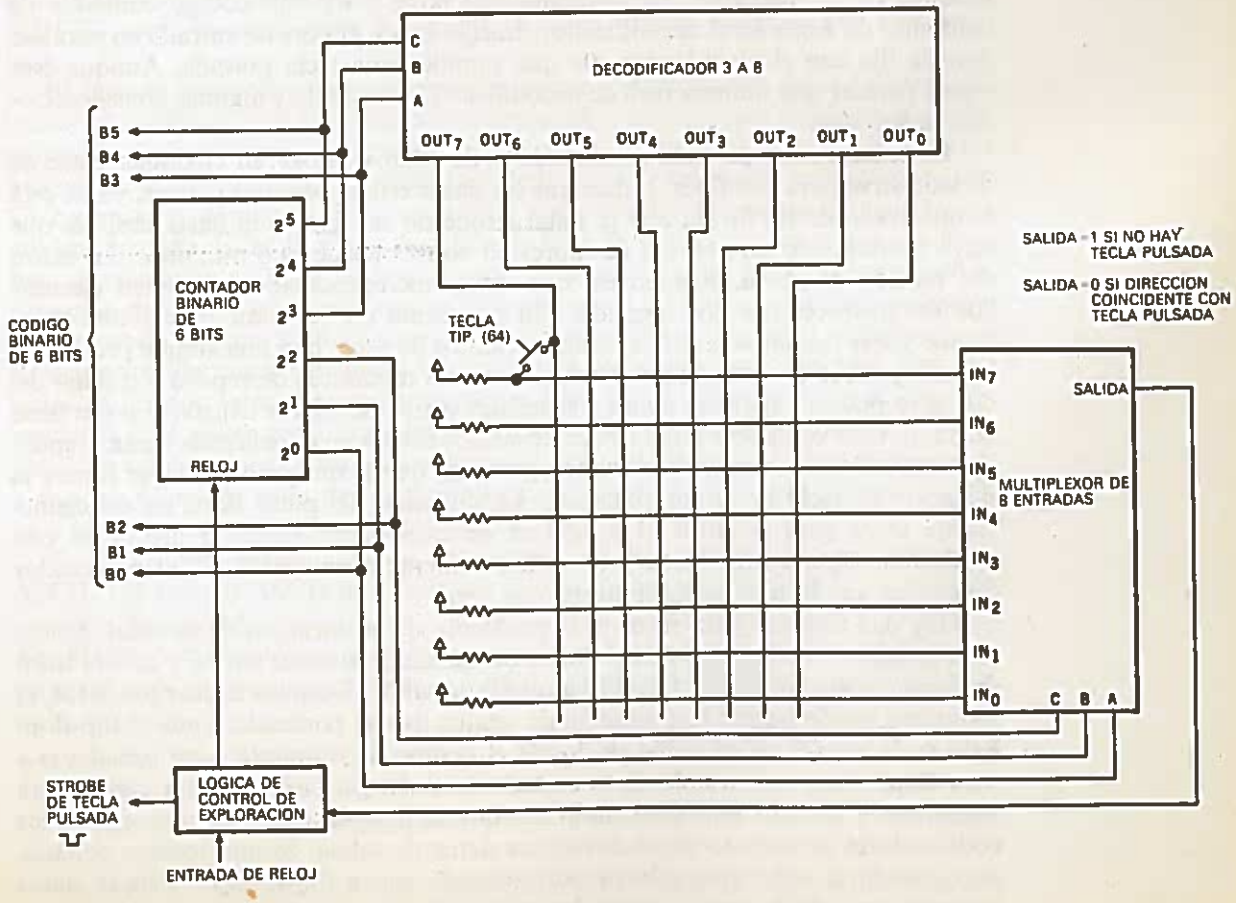


Figura 5.2 *Un explorador de tecla de matriz para un teclado de 64 teclas.*

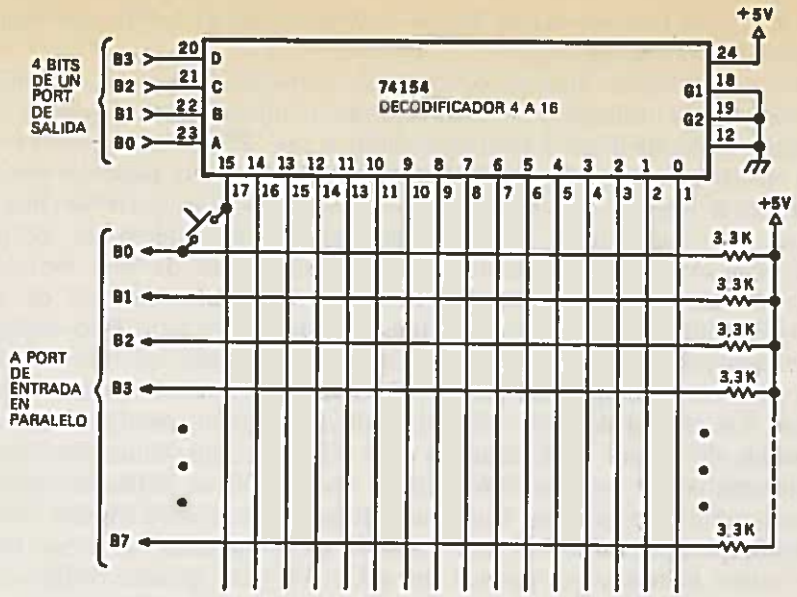


Figura 5.3 *Circuito codificador de 128 teclas controlado por software.*

de la tecla oprimida. La compatibilidad con el código ASCII es simplemente una cuestión de colocar la tecla adecuada en la dirección correcta dentro de la matriz.

Otro método adecuado de codificar está bosquejado en la figura 5.3. Esta técnica, que utiliza lógica de software para explorar la matriz, debe ser utilizada solamente cuando la velocidad de ejecución del programa de computadora no es crítica. Cuando reducimos la circuitería a un solo chip, ello implica que sea necesario un port de entrada y otro de salida. Funciona en la misma forma que se muestra en la figura 5.2. La computadora pone a «1» un código contador de columnas de 4 bits en el decodificador. Luego, busca el port de entrada en paralelo para la fila con el nivel lógico «0» que significa una tecla pulsada. Aunque esto pueda parecer una manera fácil de decodificar 128 teclas, hay algunas consideraciones de software.

La señal strobe de tecla presionada o de «datos listos» en cualquier clase de teclado sirve para dos fines: Indica que los datos están presentes y listos, y que está temporizado de tal forma que la señal strobe no sea generada hasta después que haya transcurrido un período de supresión de rebotes de tipo mecánico. La razón del retardo es obvia. Recuérdese que estos microprocesadores pueden ejecutar 200 000 instrucciones por segundo. Un programa escrito para buscar una señal strobe y leer los datos podría ejecutarse cientos de veces con una simple presión de la tecla porque el contacto rebota. Las acciones mecánicas de reposo y trabajo del contacto podrían aparecer como 100 señales strobe de «datos listos», si no se tiene cuidado. Una verdadera señal strobe de «datos listos» no es generada hasta después de tiempo de menos de 200 ns (nanosegundos), que es una velocidad que supera la duración de ciclo de la computadora. La duración del pulso debe ser suficientemente *larga* para permitir al programa de exploración captarlo incluso si está realizando alguna otra tarea y lo suficientemente *corto* para que el procesador central no vea la misma señal strobe dos veces.

Hay dos técnicas para resolver el problema de la duración de la señal strobe. Una es activar un flip-flop con el flanco de subida de la señal strobe y unir la línea de puesta a «0» (clear) del flip-flop a un bit de salida. Después de leer los datos, el programa puede borrar la condición de «datos listos» poniendo a «0» el flip-flop. Esto suele emplearse en casos en donde el tiempo de respuesta a un teclado, o a otro dispositivo, es variable. Este método también garantiza que un evento será registrado y no será omitida debido a retardos. Por supuesto, la mayoría de los codificadores de teclado no enclavan sus datos de salida. Si una tecla es soltada, aun cuando la señal strobe haya sido activada en un flip-flop, no habrán datos presentes cuando la computadora lea el teclado. Hay formas para conseguir que esto se pudiera realizar, pero implicaría un hardware adicional.

Por lo general el problema de los experimentadores es la lectura de una señal strobe dos veces, más bien que no esperar suficiente tiempo para acusar recibo de la misma. En vez de utilizar un flip-flop de hardware, la mayoría de los programadores emplean un indicador de software, que es la segunda técnica para tratar la duración de la señal strobe. Cuandó se detecta una señal strobe de tecla pulsada, el programa pone un indicador en una posición de memoria, lee los datos y luego revisa de nuevo la señal strobe. Si esta última está a nivel alto, el indicador se revisa y los datos no son leídos. Solamente cuando la señal strobe retorna a un nivel lógico «0» se pone a «0» el indicador, activando la entrada de datos para la siguiente vez.

No es fácil construir codificaciones de teclado para teclados de 64 o de 128 teclas ASCII. Es más sencillo utilizar un codificador de memoria de sólo lectura, de exploración, disponible en el comercio tal como el indicado en el Apéndice C6.

En lo que respecta al ZAP, es importante aprender a caminar antes de que corramos. La mayoría de las personas podrían considerar el ZAP como una herramienta de aprendizaje, que podría eventualmente ampliarse en un sistema de microcomputadora completamente desarrollado. Un teclado completo ASCII de 128 teclas podría ser tan caro como toda la computadora ZAP. Para reducir al mínimo los gastos y mantener las cualidades experimentales de este trabajo, un teclado limitado es recomendado como el primer nivel de expansión conveniente para una entrada hexadecimal. Con un número limitado de teclas para codificar, una circuitería TTL cableada ofrece un coste razonable frente a las costosas memorias de sólo lectura del codificador.

La figura 5.4 es un interfaz de teclado hexadecimal diseñado específicamente para el monitor de software del ZAP. Un teclado hexadecimal permite la entrada de datos e instrucciones como números hexadecimales de dos dígitos. Además de las 16 teclas numéricas, hay tres teclas de órdenes designadas «EXEC» (para ejecución), «NEXT» y «SHIFT». «EXEC» y «NEXT» serán explicadas en la sección de monitor. «SHIFT» es similar a un teclado regular y se utiliza para doblar el número de código de teclas, permitiendo, por ejemplo, SHIFT 1, SHIFT 2, etc. El significado particular de cada código se explicará más adelante.

El teclado requerido para soporte del monitor de software del ZAP tiene 19 teclas. El codificador en la figura 5.4 es explorador mixto y una salida en paralelo cableada. La codificación depende de la tecla particular pulsada. Las teclas hexagesimales, de 0 hasta F, se detectan mediante un explorador multiplexado, un CI 2 y un CI 3. A medida que cuenta CI 2, coloca secuencialmente un nivel lógico «0» en cada una de las 16 líneas de salida de CI 3. Si se pulsa alguna tecla, ese nivel bajo es retornado a CI 4 y detiene el reloj. El contador está, pues, enclavado en la dirección de la tecla particular que ha sido pulsada. La misma acción que detiene el reloj también dispara a un monoestable CI 5, que genera una señal «strobe» de la tecla presionada, mientras que el bit 7 quedara reservado para «strobe». Las tres teclas de función están directamente unidas a los bits de entrada 4, 5 y 6. Las tres secciones CI 1 sirven para amortiguar el rebote de contacto. Las teclas «EXEC» y «NEXT» están unidas de tal forma que generarán un strobe de tecla pulsada al ser activadas. Como la tecla de desplazamiento siempre se utiliza en conjunción con otra tecla, no está conectada al circuito de «strobe».

Es importante apreciar que la codificación de este circuito de 19 teclas no es ASCII. Un teclado ASCII no puede ser utilizado directamente con el monitor de software bosquejado en este libro, a menos que utilice solamente las teclas del ASCII que corresponden a la codificación de la figura 5.4, o escribir de nuevo el monitor de software para aceptar el ASCII mejor que el código binario de cada tecla.

II. ADICION DE UNA PRESENTACION VISUAL

Una vez se ha añadido un teclado al ZAP, estamos preparados para un desarrollo de programa. El otro ingrediente clave es una presentación visual, que permite al programador examinar las sentencias de instrucción necesarias y los

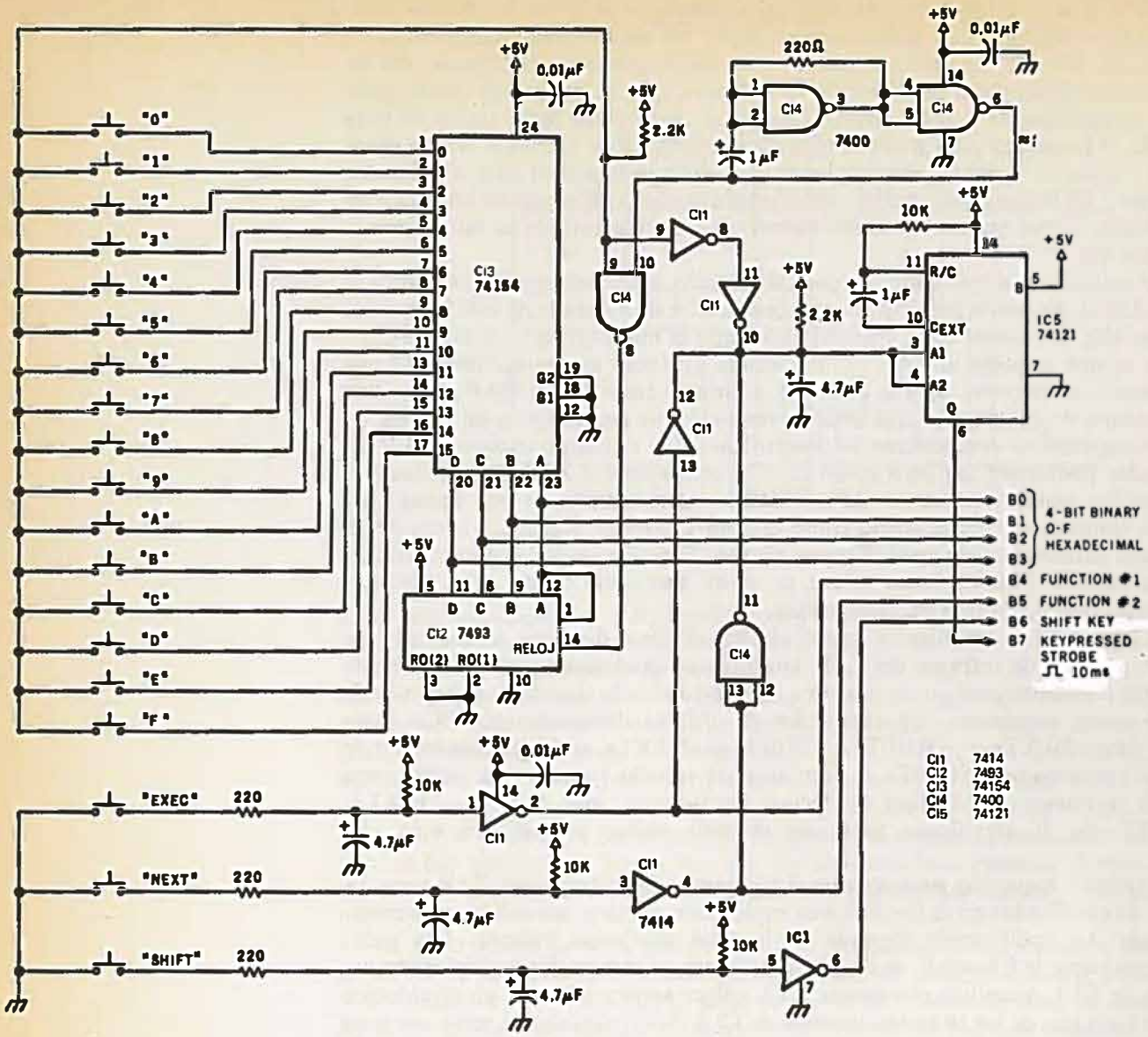


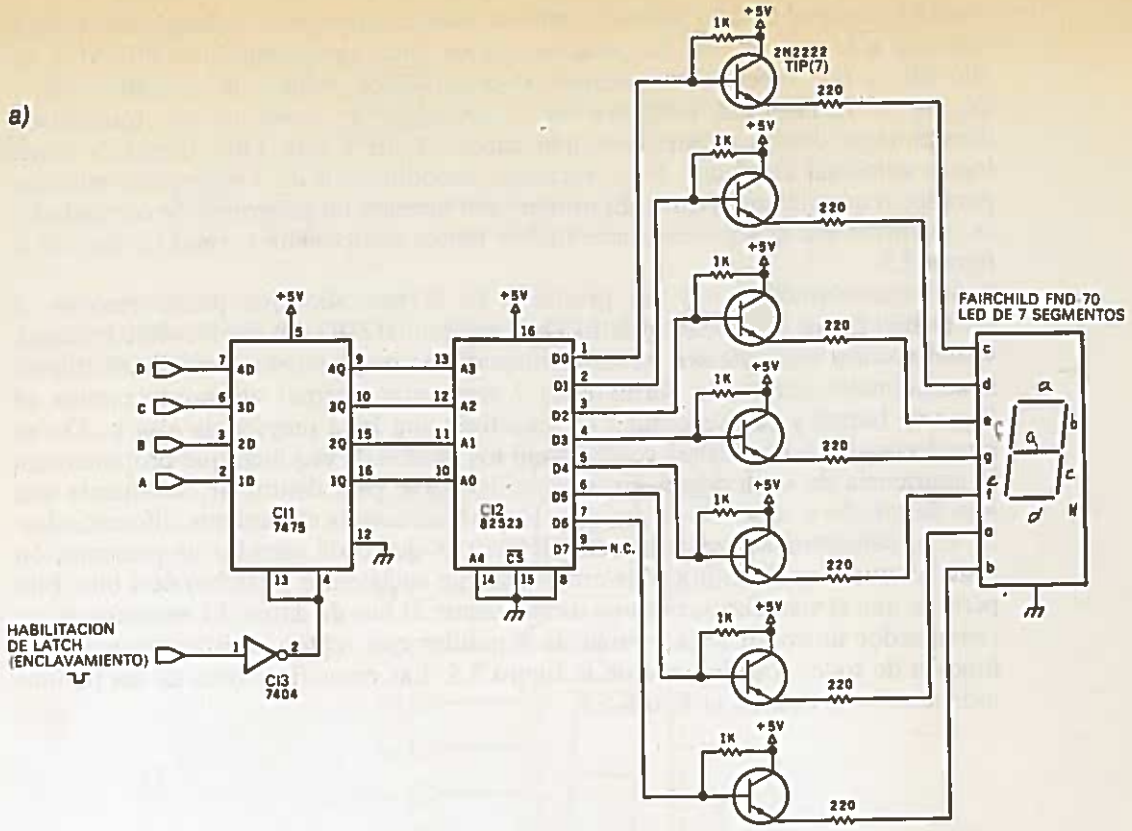
Figura 5.4 *Un interfaz de teclado hexadecimal.*

datos. La configuración menos costosa en un display LED, preferiblemente hexadecimal, puesto que el monitor de software está escrito en esa forma. Para el modelo octal no se ha incluido una visualización octal.

Las visualizaciones hexadecimales pueden parecer triviales en un sistema de computadora caro, pero a veces estas adiciones poco útiles son las que permiten corregir más fácilmente el programa. No pretendo que deba sustituir a un TRC, pero es una herramienta útil cuando se corrige un programa y una necesidad para la utilización del monitor del ZAP. Nunca reemplazará a un seguidor paso a paso o a un programa monitor de punto de ruptura, pero es muy interesante para presentar datos de E/S, o teclado, rápidamente con una sola instrucción de salida.

Hay muchas formas de presentar hexadecimal en un LED de siete segmentos. La figura 5.5 es un ejemplo del método de «fuerza bruta» utilizando una PROM como un decodificador hexadecimal. (Un método de programar el 82S23 se describió en el número de noviembre de 1975 de la revista *BYTE* titulado «Un programador versátil de memoria de sólo lectura», si escogió utilizar este circuito.)

a)



b)

CODIGO DE ENTRADA				PROGRAMA 82S23								VISUALIZADOR (DISPLAY) DE 7 SEGMENTOS
D	C	B	A	D7	D6	D5	D4	D3	D2	D1	D0	
0	0	0	0	0	1	1	1	0	1	1	1	0
0	0	0	1	0	1	0	0	0	0	0	1	1
0	0	1	0	0	1	1	0	1	1	1	0	2
0	0	1	1	0	1	1	0	1	0	1	1	3
0	1	0	0	0	1	0	1	1	0	0	1	4
0	1	0	1	0	0	1	1	1	0	1	1	5
0	1	1	0	0	0	1	1	1	1	1	1	6
0	1	1	1	0	1	1	0	0	0	0	1	7
1	0	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	0	1	1	1	1	0	0	1	9
1	0	1	0	0	1	1	1	1	1	0	1	A
1	0	1	1	0	0	0	1	1	1	1	1	b
1	1	0	0	0	0	1	1	0	1	1	0	C
1	1	0	1	0	1	0	0	1	1	1	1	d
1	1	1	0	0	0	1	1	1	1	1	0	E
1	1	1	1	0	0	1	1	1	1	0	0	F

Figura 5.5 *Un método posible para un latch/decodificador/excitador hexadecimal, es el empleo de un LED de 7 segmentos estándar.*

a) Este circuito completo sería necesario para sustituir un HP7340. $\overline{CS}$ en el 82523 puede realizar la función de borrado.

b) El programa para el 82523 (CI 2).

Sin embargo, en este procedimiento se utiliza un número excesivo de componentes y a la mayoría de las personas no les gusta programar una PROM. Una alternativa que tenemos es permitir al computador realizar la decodificación y excitar el visualizador (display) de 7 segmentos a través de los transistores directamente desde un port de salida enclavado de 8 bits. Otra forma es poner lógica adicional alrededor de un excitador decodificador de 7 segmentos estándar para los requerimientos extra. El primer caso necesita un programa de computadora, mientras que el segundo puede incluir tantos componentes como los hay en la figura 5.5.

Afortunadamente, hay un producto en el mercado que puede resolver el problema. Es un visualizador de LEDs hexadecimal HP7340 (de Hewlett Packard; visualizadores equivalentes pueden obtenerse de otros fabricantes). Estos dígitos hexadecimales parten del formato de 7 segmentos normal utilizando puntos en lugar de barras y con capacidad de visualizar una letra mayúscula «B» y «D» en hexadecimal. Ello se realiza controlando los puntos de esquina, que proporcionan la apariencia de «redondeo». Esta capacidad sirve para distinguir claramente una «B» de un «8» o una «D» de un «0». Hay 16 caracteres claramente diferenciados.

Una característica adicional del HP7340 es que cada circuito de presentación visual contiene un decodificador/excitador y un enclavador («latch») de 4 bits. Ello permite que el visualizador se una directamente al bus de datos. El resultado es un visualizador único en hexadecimal de 8 patillas que realiza satisfactoriamente la función de todos los circuitos de la figura 5.5. Las especificaciones de las patillas individuales se dan en la figura 5.6.

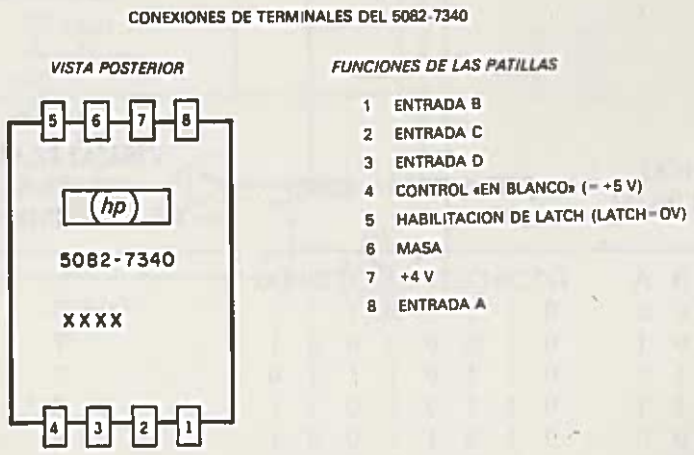


Figura 5.6 *La disposición de terminales y las funciones para el visualizador HP7340 de BCD a hexadecimal. Visualizadores similares se producen por Dialite y Texas Instruments.*

Las figuras 5.7 y 5.8 ilustran cómo puede configurarse el HP7340 para funcionar como un port de salida hexadecimal de 2 dígitos o un port octal de 3 dígitos. No se requiere un enclavador («latch») de 8 bits porque ya contiene uno. Los circuitos HP7340 pueden unirse al bus de datos tan simplemente como cualquier otro port de salida en paralelo y se les aplica la función «strobe» a partir del decodificador de selección de chips que se describió anteriormente en la sección sobre decodificación de E/S.

Para utilizar adecuadamente el monitor de software, se necesitan 6 visualizadores hexadecimales (separados en 3 visualizadores de un solo octeto). Tres octetos se precisan para visualizar una dirección particular H y L y el contenido de datos de esa posición. Los 6 visualizadores hexadecimales deben tener las señales «strobe» decodificadas siguientes:

Port de salida núm.	Línea de lógica	Parámetro de visualización	CI núm.
5	$\overline{DS5WR}$	Campo direcciones MSD	30, 31
6	$\overline{DS6WR}$	Campo direcciones LSD	28, 29
7	$\overline{DS7WR}$	Campo de datos	26, 27

MSD—Dígito más significativo
LSD—Dígito menos significativo

Una descripción más completa de cada función de presentación visual se describe dentro de la sección del monitor y un esquema completo que ilustra cómo los 6 visualizadores están unidos al bus de datos se da en la figura 5.9.

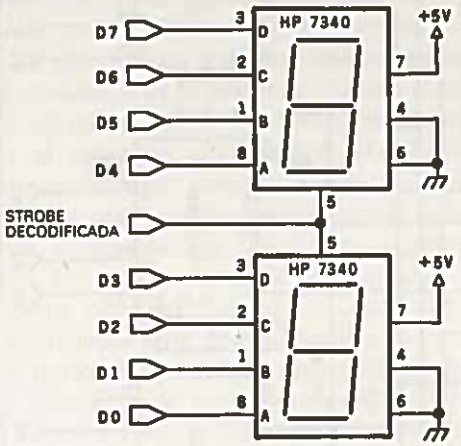


Figura 5.7 *Un visualizador hexadecimal de enclavador/decodificador/excitador HP7340.*

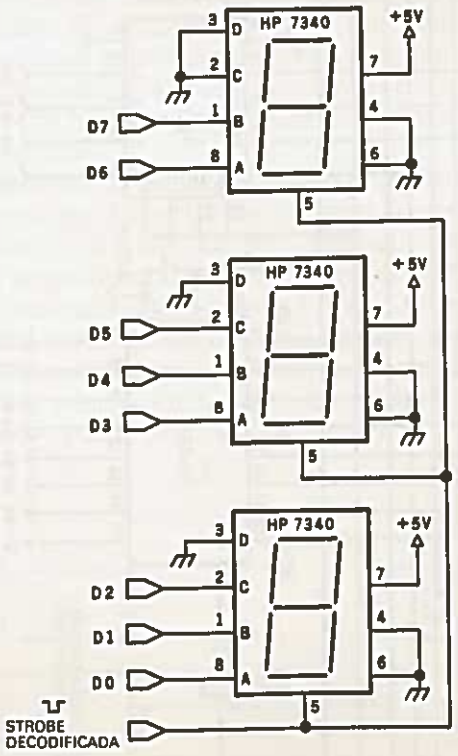


Figura 5.8 *Un visualizador octal de enclavador/decodificador/excitador HP7340. El HP5082-7300 puede sustituirse por el HP5082-7340 en aplicaciones de visualización octal. El HP7300 visualiza solamente caracteres numéricos.*

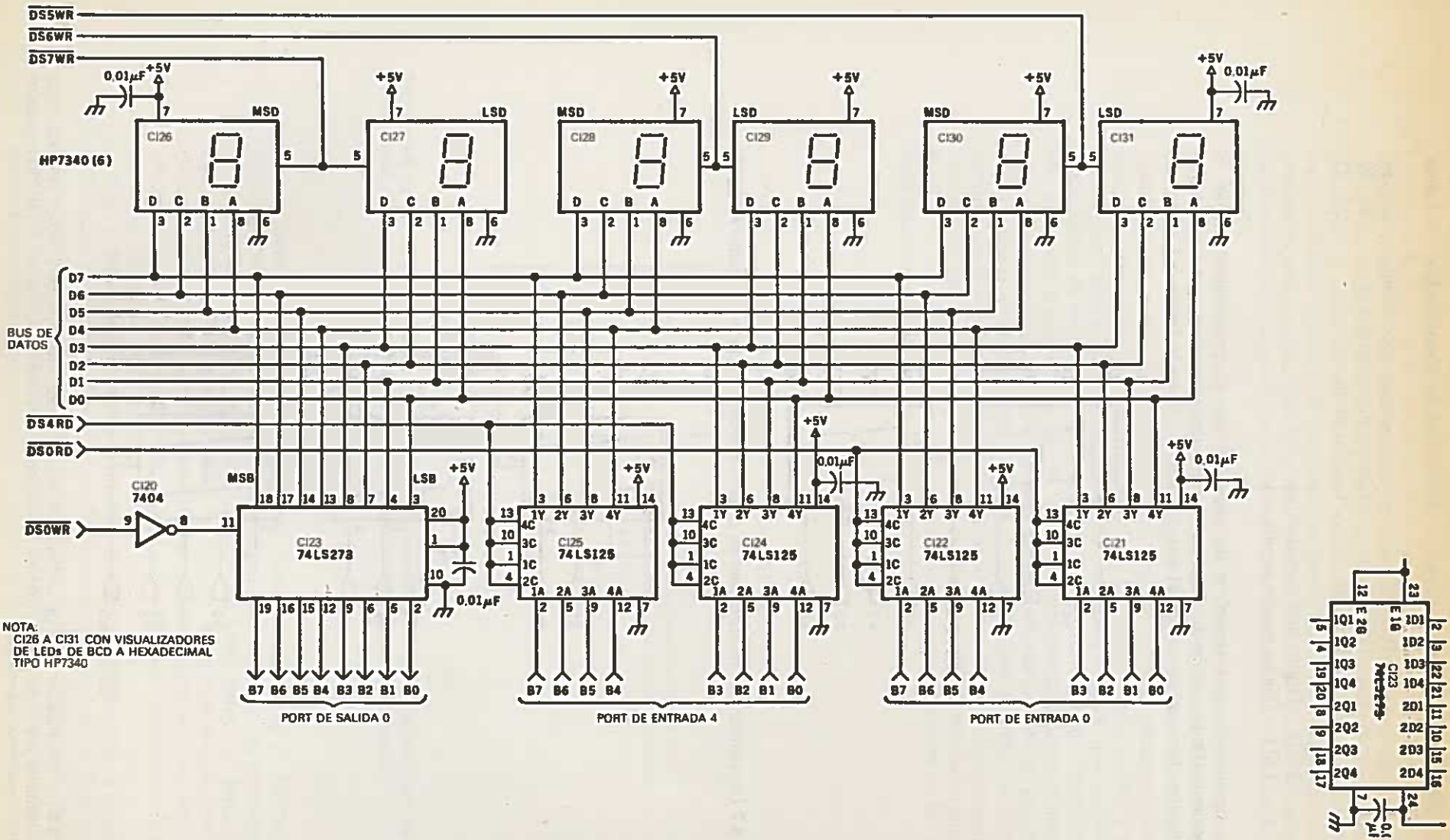


Figura 5.9 *Un esquema del visualizador hexadecimal completo de LEDs.*

III. INTERFAZ EN SERIE

Unos medios que posibiliten la comunicación en serie no son absolutamente necesarios para hacer funcionar el ZAP, aunque el monitor de software suministrado en este libro soporta un interfaz en serie.

En primer lugar, hablemos sobre los conceptos antes de proseguir con los detalles del diseño. ¿Por qué ZAP necesitaría comunicarse? Cuando examinemos el interfaz de casete en serie, comprenderemos que hay más ventajas que las que se manifiestan actualmente. Si se tiene prevista una futura expansión o si se han de añadir periféricos comercialmente disponibles tales como TRC o impresora, su interfaz probablemente será en serie.

Esta última aseveración es significativa. Hay que percatarse de que no dije nada acerca de la comunicación con otra computadora. Aunque la comunicación con otra computadora a través de líneas telefónicas requiere un enlace en serie, los periféricos normales, tales como TRC e impresoras, también se comunican en serie. Por consiguiente, al diseñar un port en serie para acoplar una impresora, también adquirimos los medios para comunicarnos con otra computadora.

La comunicación es simplemente la transferencia de información desde un dispositivo a otro. En el caso de una unidad de visualización de TRC, la computadora envía información de caracteres para visualización en pantalla, mientras que el teclado transmite la entrada del usuario a la computadora. Cada extremo de la línea de comunicación de dúplex compuesto debe tener un transmisor y un receptor. En ambos casos, la información que se transfiere son datos ASCII probablemente constituidos por un código de 7 bits y, en algunos casos, un bit de paridad adicional para comprobación de errores. Estos datos de 7 bits (ignorando el bit de paridad) aparecerán en las líneas de un port en paralelo. Estas 7 líneas más una referencia de masa y una señal «strobe» (recuérdese que hemos de comunicar al receptor cuándo los datos son válidos) pueden llevarse a la entrada del TRC. Conservándole como una línea dedicada desde la computadora al TRC, ahora deseamos tener una línea similar entre la salida del teclado y un port en paralelo de 8 bits en la computadora. Ello exige 9 líneas adicionales. Para complicar todavía más el asunto, separemos el terminal y la computadora en una distancia de 100 a 120 metros, como pudiera suceder en algunos sistemas de computadoras comerciales. El resultado es que 120 metros de cable de 18 conductores (17 si se combinan referencias de masa) costarán más que el terminal. También ha de considerarse que la salida en paralelo TTL no debe emplearse para excitar líneas más largas que 6 metros, sin buffers/excitadores especiales; de no ser así, podrían producirse errores.

La solución a este problema de cableado costoso es utilizar la comunicación en serie más bien que en paralelo. Los datos en paralelo se convierten en serie y se envían, un bit cada vez, a un hilo de par trenzado único. Si se necesitan buffers/excitadores para largas distancias, menos se requieren con el procedimiento en serie. Los bits de comienzo y de parada («start» y «stop») especialmente codificados, e incluidos en la transmisión en serie, indican al receptor que se están enviando datos válidos. Para el ejemplo anterior, solamente dos pares de hilos se necesitan para realizar una interacción de «dúplex completo» (ver figura 5.10). En el modo de «semidúplex» puede reducirse a un par trenzado único, pero la

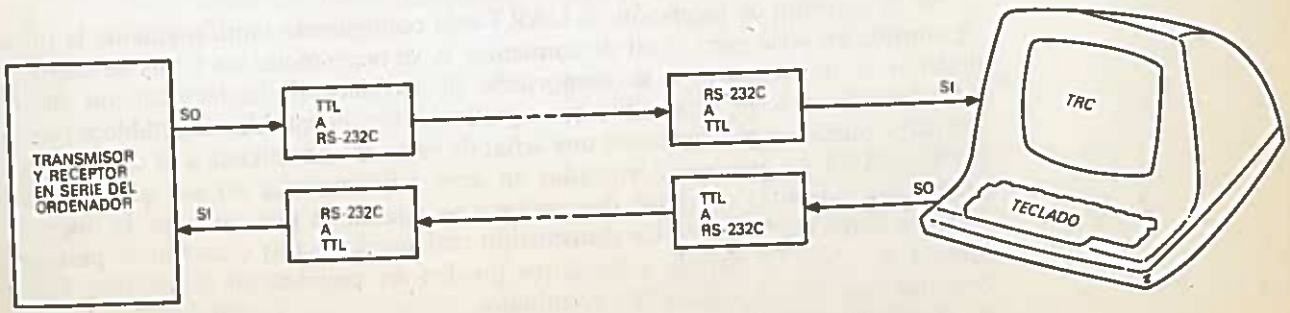


Figura 5.10 *Diagrama en bloques de un enlace de comunicación de dúplex completo RS-232C.*

sincronización de la línea de comunicación compartida es más complicada. Todas las referencias de transmisión en serie que haremos se limitarán al funcionamiento en dúplex completo.

Ahora que hemos convenido en que la comunicación debe ser en serie, ¿cómo realizamos la conversión paralelo a serie? La respuesta es un dispositivo denominado UART (transmisor/receptor asíncrono universal). En el Apéndice C7 se da la información de especificación para el UART SMC COM2017, que tiene una función equivalente al AY-5-1013A (General Instruments). Para reducir al mínimo las necesidades de alimentación, puede sustituirse por una sola de +5 V, AY-3-1015 o TR1602 (Western Digital), en la forma que se indica. El único cambio de la hoja de especificación es que el terminal núm. 12 ya no está unido a −12 V.

Una estructura interna del UART está constituida por un transmisor de paralelo a serie y un receptor de serie a paralelo independientes unidos por terminales de programación comunes. Ello significa que las dos secciones del UART pueden utilizarse con independencia, siempre que cumplan con el mismo formato de bits que sea seleccionable por software en el chip o cableado.

La transmisión desde la computadora al TRC se realiza de forma asíncrona y sólo en un sentido. Análogamente, la computadora recibe datos directamente desde el teclado a través de una línea dedicada. En lo que respecta a la computadora, después de la reconversión a paralelo en el UART, este dispositivo de entrada estará comunicando datos en paralelo.

La transmisión de datos real sigue el formato en serie asíncrono ilustrado en la figura 5.11. Con el empleo del teclado como ejemplo, cuando no están transmitiendo datos, la línea de datos se sitúa en una marca de trabajo (o nivel «1») a la espera de una señal «strobe» por pulsación de tecla. Esta última señal es un impulso positivo de 1 a 5 ms (puede ser tan corto como 200 ns), que indica que se ha pulsado una tecla y que un código ASCII de esa tecla está disponible para transmisión. La señal «strobe», que está unida a la «strobe» de datos del UART, hace que los datos en ASCII se carguen en una memoria intermedia en paralelo e inicia el ciclo de transmisión del UART. La salida en serie hará entonces una transición desde «1» a «0». Este bit de comienzo de marca a «0» tiene una longitud de un período de reloj e indica el principio de una palabra transmitida en serie. Después del bit de comienzo, siguen hasta 8 bits de datos, teniendo cada bit de datos la longitud de un período de reloj. A la conclusión de los bits de datos, los bits de paridad y de parada se hacen salir por el UART para indicar el final de transmisión. Si se pulsa otra tecla, el proceso se repite a sí mismo.

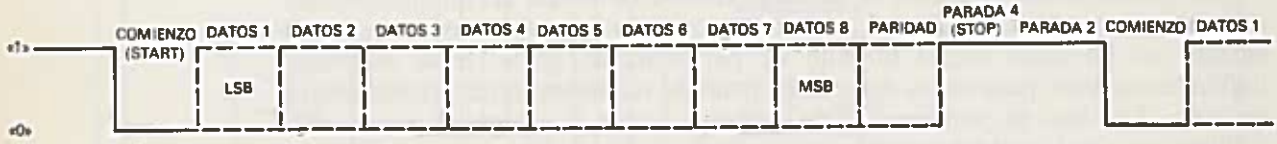


Figura 5.11 *Un octeto de datos único según se transmite en el formato en serie asíncrono.*

En el extremo de recepción, el UART está controlando continuamente la línea de entrada en serie para el bit de comienzo. A su ocurrencia, los 8 bits de datos se deslizan a un registro y se comprueba la paridad. A la terminación de la introducción en serie, una salida que significa datos disponibles se establece por el UART y puede emplearse como una señal de «strobe» de entrada a la computadora. El UART no procesará entradas en serie adicionales a no ser que se acuse recibo del indicador de datos disponible y se seleccione por «strobe» la línea de reset de datos disponibles. La transmisión real puede incluir o excluir la paridad, tener 1 ó 2 bits de parada y los datos pueden ser palabras de 5 a 8 bits. Estas opciones son seleccionables por terminales.

A continuación se da una descripción de las funciones de los terminales (pines) para el AY-5-1013, COM2017 o AY-3-1015.

Term. núm.	NOMBRE	SIM-BOLO	FUNCION
1	V_{CC} Fuente de alimentación	V_{CC}	Alimentación de +5V.
2	V_{GG} Fuente de alimentación	V_{GG}	Alimentación de −12V (no conectada en AY-3-1015).
3	Masa	GND	Masa.
4	Habilitación datos recibidos (Received Data Enable)	$\overline{RDE}$	Un nivel lógico «0» en la línea de habilitación del receptor coloca los datos recibidos en las líneas de salida.
5 hasta 12	Bits de datos recibidos (Received Data Bits)	RD8 hasta RD1	Estas son las 8 líneas de salidas de datos. Los caracteres recibidos están justificados a la derecha; el LSB siempre aparece en RD1. Estas líneas tienen salidas de 3 estados.
13	Error de paridad (Parity Error)	PE	Esta línea pasa a un nivel lógico «1» si la paridad de los caracteres recibidos no coincide con la paridad seleccionada. Tres estados.
14	Error de encuadre (Framing Error)	FE	Esta línea pasa a un nivel lógico «1» si el carácter recibido no tiene un bit de parada válido. Tres estados.
15	Sobreejecución (Over-run)	OR	Esta línea pasa a un nivel lógico «1» si el último carácter que ha llegado no ha sido leído (la línea DAV no se ha puesto a cero) antes de que el carácter se transfiera al registro de retención del receptor. Tres estados.
16	Habilitación de palabras de estado (Status Word Enable)	$\overline{SWE}$	Un nivel lógico «0» en esta línea pone los bits de palabra de estado (PE, FE, OR, DAV, TMBT) en las líneas de salida. Tres estados.
17	Reloj receptor (Receiver Clock)	RCP	Esta línea debe tener como entrada un reloj cuya frecuencia sea 16 veces (16×) la velocidad deseada de transmisión de datos del receptor.
18	Reinicialización datos disponibles (Reset Data Available)	$\overline{RDAV}$	Un nivel lógico «0» reinicializará la línea DAV.
19	Datos disponibles (Data Available)	DAV	Esta línea pasa a un nivel lógico «1» cuándo un carácter completo se haya recibido y transferido al registro de retención del receptor. Tres estados.
20	Entrada en serie (Serial Input)	SI	Esta línea acepta la corriente de entrada de bits en serie. Para la iniciación de recepción de datos se requiere

			una transición de marca de trabajo (nivel lógico «1») a espacio de reposo (nivel lógico «0»).
21	Reinicialización externa (External Reset)	XR	Reinicializa registros de desplazamiento. Pone SO, EOC y TBMT a un nivel lógico «1». Pone a «0» los indicadores de error y DAV. Borra la memoria intermedia de datos de entrada. Debe unirse al nivel lógico «0» cuando no está en uso.
22	Memoria intermedia del transmisor vacía (Transmitter Buffer Empty)	TBMT	El indicador de estado vacío de la memoria intermedia del transmisor pasa a nivel lógico «1» cuando el registro de retención de bits de datos pueda cargarse con otro carácter. Tres estados.
23	Strobe de datos (Data Strobe)	$\overline{DS}$	Una señal «strobe» en esta línea introducirá los bits de datos en el registro de retención correspondiente. La transmisión inicial de datos se inicia por el flanco de subida de $\overline{DS}$. Los datos deben ser estables durante la señal «strobe» completa.
24	Final de carácter (End of Character)	EOC	Esta línea pasa al nivel lógico «1» cada vez que se transmite un carácter completo. Se mantiene en este nivel hasta el comienzo de la transmisión del siguiente carácter.
25	Salida en serie (Serial Output)	SO	Esta línea proporcionará en serie, bit a bit, el carácter transmitido completo. Se mantendrá al nivel lógico «1» cuando no se esté transmitiendo ningún dato.
26 a 33	Entrada de bits de datos (Data Bits Inputs)	BD1 a BD8	Hay hasta ocho líneas de entrada de bits de datos disponibles.
34	Strobe de control (Control Strobe)	CS	Un nivel lógico «1» en este terminal introducirá los bits de control (EPS, NB1, NB2, TSB, NP) en el registro de retención de bits de control. Esta línea puede ser objeto de «strobe» o de cableado a un nivel lógico «1».
35	No paridad (No Parity)	NP	Un nivel lógico «1» en este terminal eliminará el bit de paridad del carácter transmitido y recibido (ninguna indicación de PE). Los bits de parada («stops») seguirán inmediatamente al último

			bit de datos. Si no se utiliza, este terminal debe unirse a un nivel lógico «0».
36	Número de bits de parada (Number of Stop Bits)	TSB	Seleccionará el número de bits de parada, uno o dos, a añadir inmediatamente después del bit de paridad. Un nivel lógico «0» insertará 2 bits de parada. Un nivel lógico «1» insertará un bit de parada.
37 38	Número de bits/caracteres (Number of Bits/Characters)	NB2, NB1	Se decodificarán internamente para seleccionar 5, 6, 7 u 8 bits de datos/caracteres.

NB2	NB1	Bits/carácter
0	0	5
0	1	6
1	0	7
1	1	8

39	Selección paridad impar/par (Odd/Even Parity Select)	EPS	El nivel lógico en este terminal selecciona el tipo de paridad que se añadirá inmediatamente después de los bits de datos. También determina la paridad que será comprobada por el receptor. Un nivel lógico «0» insertará paridad impar y un nivel lógico «1» insertará paridad par.
40	Reloj transmisor (Transmitter Clock)	TCP	Esta línea debe tener como entrada una señal de reloj cuya frecuencia es 16 veces la velocidad deseada de transmisión de datos del propio transmisor.

En la figura 5.12 se muestra la configuración final del interfaz en serie. Puesto que un UART es un dispositivo triestado, puede unirse directamente al bus de datos. Los datos son objeto de escritura o lectura en el mismo, en la forma de 8 bits en paralelo como cualquier otra manipulación de ports (puertos) de E/S. Para la computadora, el UART aparece como registros de una salida y dos entradas: estado, datos transmitidos y datos recibidos. Como con todas las manipulaciones de bus de datos, las transferencias de datos están sincronizadas mediante «strobes» decodificadas. El monitor de software del ZAP utiliza tres direcciones de ports para coordinar el hardware y el software. Para ser compatibles, deben cablearse como sigue:

Port núm.	Línea de lógica	Señal
02 INPUT (Entrada)	$\overline{\text{DS2RD}}$	READ DATA (Datos de lectura)
03 INPUT (Entrada)	$\overline{\text{DS3RD}}$	READ STATUS (Estado de lectura)
02 OUTPUT (Salida)	$\overline{\text{DS2WR}}$	WRITE DATA (Datos de escritura)

El enfoque primario de este capítulo es la sección de hardware del interfaz en serie. Cuando se conecta directamente al bus de datos en esta manera, no hay forma de operar el UART, excepto bajo un control de programa. La explicación del protocolo y el significado de cada registro del UART puede encontrarse en la sección acerca del monitor de ZAP.

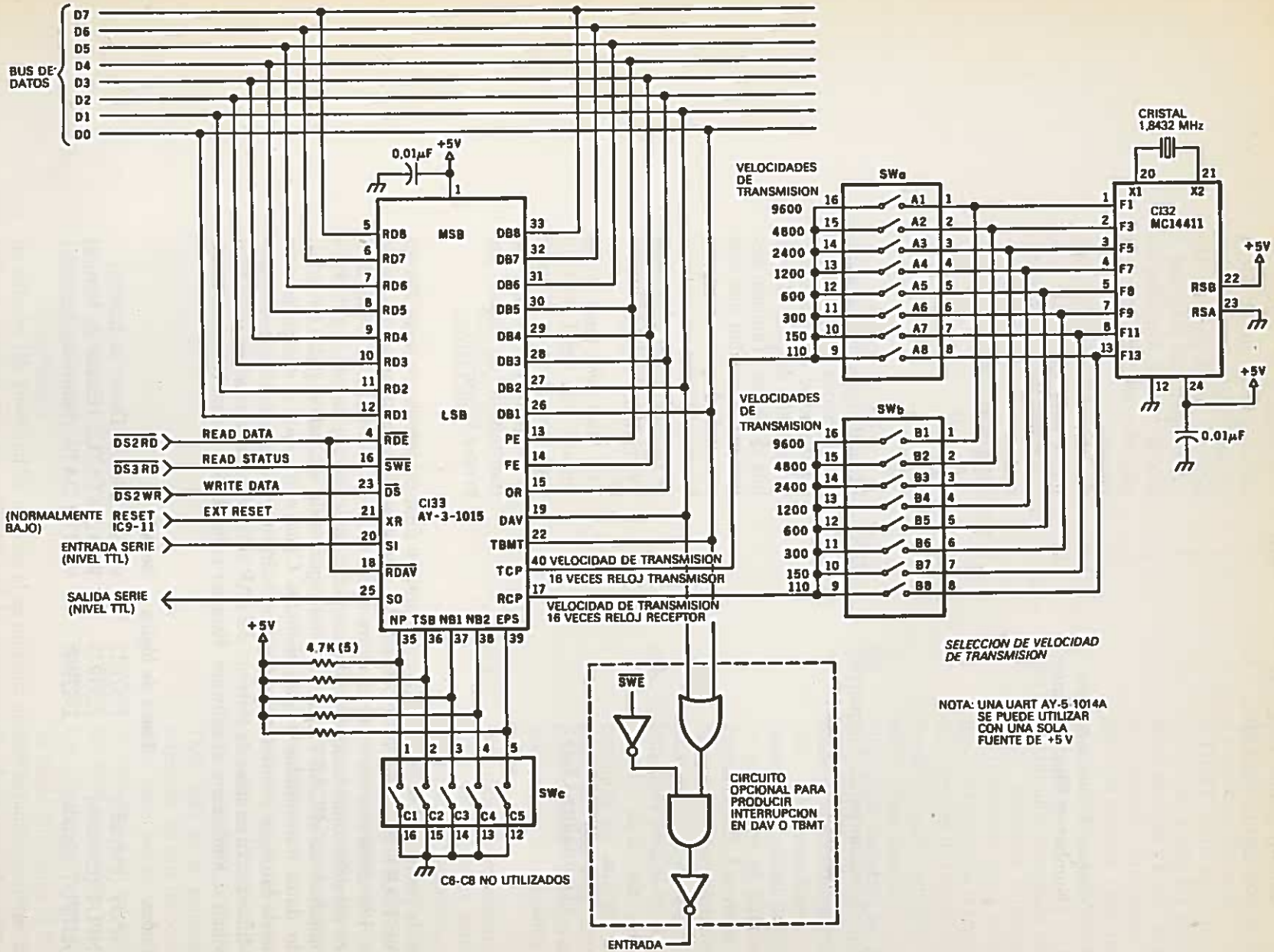


Figura 5.12 *Configuración final del interfaz en serie.*

Hay dos restantes consideraciones del hardware: la velocidad de transmisión de datos y el nivel de las señales de transmisión. La velocidad de transmisión de los datos puede ser expresada como bits por segundo y se refiere a la velocidad de transmisión a lo largo del par trenzado. Ha de tenerse presente que a una velocidad de datos más baja solamente 8 de 11 bits de cada palabra transmitida son datos; son utilizados un bit de comienzo y 2 bits de parada. Aunque cualquier frecuencia de transmisión puede ser establecida en un UART, ajustando la frecuencia del reloj hay 8 velocidades de transmisión asíncrona normal utilizadas con frecuencia:

110 bps
150 bps
300 bps
600 bps
1200 bps
2400 bps
4800 bps
9600 bps

Utilizando un chip especial de generador de velocidad de transferencia de datos y una red selectora de interruptores, como la que se muestra en la figura 5.12, el ZAP puede admitir cualquiera de estas frecuencias específicas. En funcionamiento normal, la mayoría de los teletipos funcionan a 110 bps, las impresoras, tales como DECwriter II, a 300 bps, los modems de teléfono acústico a 300 bps, y hay terminales de vídeo que van desde los 1200 hasta los 19 200 bps. Como puede ver en teoría, podemos comunicarnos con todos ellos.

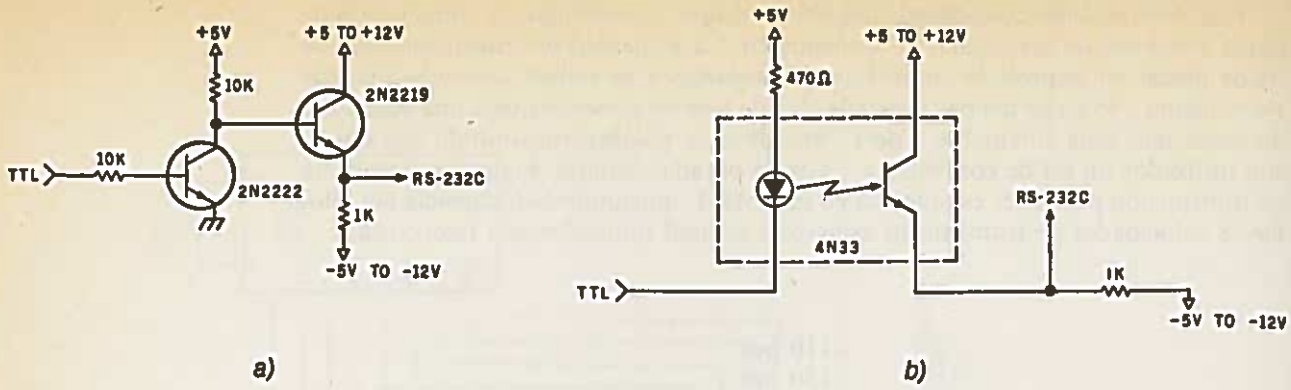
La velocidad de transmisión es solamente parte de los prerrequisitos de la intercomunicación. Una computadora podría ser de lógica de nivel TTL mientras que un periférico utilizara 15 V CMOS. Ellos serían completamente incompatibles. En consecuencia, es necesario tener una norma adicional que gobierne el nivel de señal de las transmisiones. La norma más ampliamente aceptada y generalmente utilizada es la EIA RS-232C.

Aun cuando los niveles TTL podrían ser usados para comunicación, no son apropiados para transportar señales a más de 3 a 6 metros. El problema se origina por el hecho de que solamente 2 V separa un nivel lógico «1» ó «0» más bien que las capacidades de control o de velocidad. Con sólo 2 V de inmunidad para los ruidos, la comunicación podría ser susceptible a interferencias de motores y de interruptores.

Un comité industrial acordó un interfaz normalizado para resolver este problema, así como para sugerir normas para la industria. la EIA RS-232C se utiliza por los equipos modem. Esta especificación se aplica no sólo a las tensiones específicas asignadas al nivel lógico 0 y 1, sino también al tipo de enchufe, asignaciones de terminales, impedancias de fuente y de carga, así como a una variedad de otras funciones afines.

Los niveles de señal del RS-232C son bipolares y utilizan una tensión negativa entre -3 y -15 V para representar un nivel lógico «1» y una tensión positiva de 3 a 15 V para representar un nivel lógico «0». La región entre -3 V y $+3$ V ayuda a nuestra inmunidad contra ruidos y es una región «muerta». Aun cuando ± 15 V podrían proporcionar una transmisión óptima, $+3$ V y -7 V son también aceptables. Sin embargo, trátese de mantener niveles bipolares iguales en distancias largas.

La computadora básica ZAP requiere alimentaciones de $+12$, $+5$ y -12 V (para la memoria EPROM se necesita -5 V y se obtiene de la fuente de -12 V) para funcionar. Podemos utilizar las alimentaciones positiva y negativa para generar niveles de tensión del RS-232C en varias formas. La figura 5.13 ilustra algunos controladores RS-232C, y la figura 5.14 muestra una pareja de circuitos receptores. Uno de cada selección tendrían que ser conectados a los terminales de E/S del UART para que se tenga una compatibilidad total con el RS-232C.



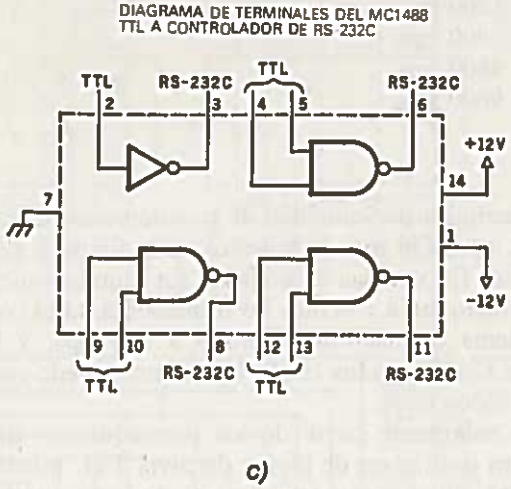


Figura 5.13 *TTL a controladores de RS-232C.*
a) Utilizando dos transistores como un desplazador.
b) Utilizando un opto-aislador como desplazador de nivel.
c) Utilizando un controlador de línea RS-232C.

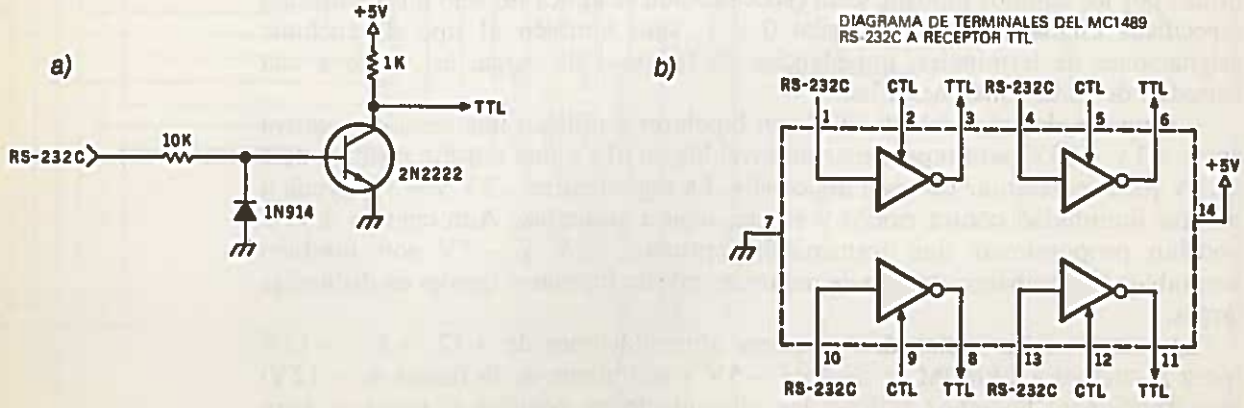


Figura 5.14 *RS-232C a receptores TTL.*
a) Utilizando un transistor.
b) Utilizando un receptor de línea RS-232C normal.

IV. INTERFAZ DE MEMORIA DE CASETES

El último, pero no de menor importancia, de los perfeccionamientos que podríamos añadir al ZAP es un interfaz de casetes. Con el teclado y el display, el operador será capaz de escribir algunos programas complicados, pero, al menos que sean transferidos a la memoria de sólo lectura, se perderán cuando se desconecta la alimentación. Por supuesto, la alimentación de la computadora puede dejarse aplicada constantemente. Pero qué pasa si se quiere desarrollar un segundo programa que deba ocupar el mismo espacio de dirección de memoria. La solución ideal es poseer algún medio que temporalmente almacene grandes bloques de memoria.

En sistemas de computadoras grandes, se consigue mediante un disco rígido y sistemas de cintas magnéticas de 9 pistas. Estos medios de alto volumen y alta velocidad se salen del objetivo de la computadora personal prevista, pero su valor en sistemas grandes es evidente. Una alternativa de bajo coste y menos prestaciones es un sistema de memoria (almacenaje) de casete de audio.

En general, un interfaz de almacenaje de casete consiste en tres subsistemas principales: Un transmisor/receptor en serie, un ensamblaje de hardware que convierte los datos en serie TTL de forma que sean compatibles con el casete de audio y un programa de aplicación que está al corriente de lo que está saliendo a la cinta y que puede cargarle en el lugar adecuado. La configuración básica es ilustrada en el diagrama de bloques de la figura 5.15.

La sección de transmisor/receptor en serie no es nada más que el interfaz en serie del UART que ya hemos agregado. Se comunica mediante un RS-232C, con los convertidores MC-1488 y 89 en sus líneas en serie. Sin embargo, si une un interfaz de casete a estas líneas, podría doblarse como un dispositivo de almacenaje. Un beneficio adicional es que los datos en serie generados por el UART ofrecerán alguna compatibilidad entre sistemas de computación personal; velocidades de transmisión de datos normales y un protocolo de comunicación en serie normal lo promocionarán.

La salida del UART es TTL. Incluso con el controlador RS-232C, la salida lógica sigue siendo de un nivel de c.c. Como las grabadoras de audio no pueden registrar c.c, la salida del UART debe ser convertida de alguna manera. La solución es FSK (modulación por desplazamiento de frecuencia). La salida TTL procedente del UART se convierte en tonos de audio. Una frecuencia representa un nivel lógico «0» y otra representa un nivel lógico «1».

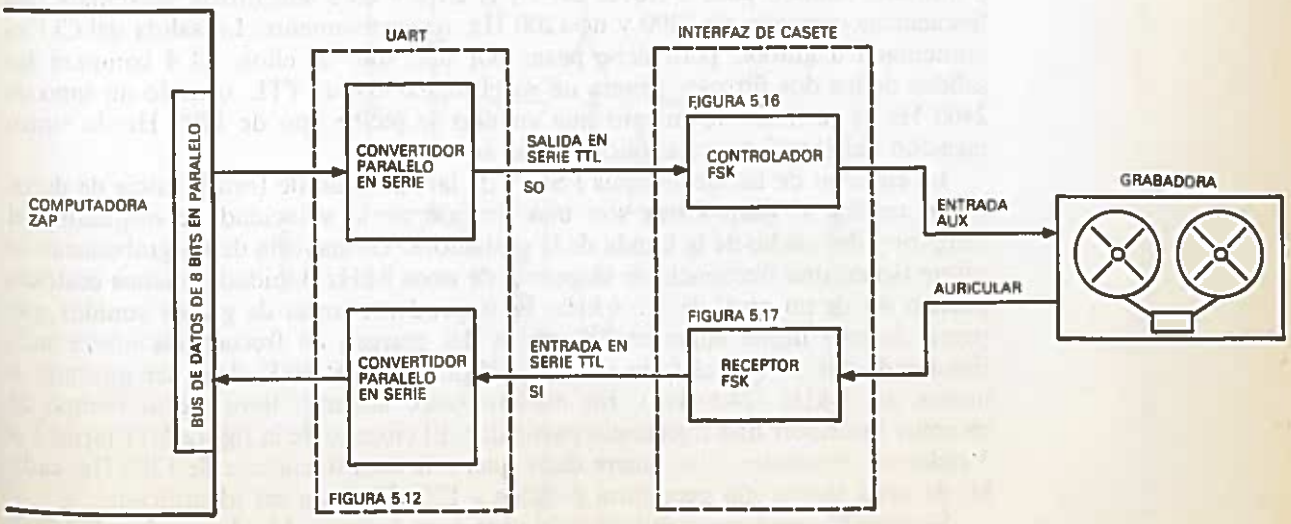


Figura 5.15 *Un diagrama de bloques de un sistema de almacenamiento de casete de audio.*

La figura 5.16 muestra un circuito que producirá tonos de audio modulados por desplazamiento de frecuencia. Una frecuencia de referencia de 4800 Hz se obtiene por un generador de velocidad de transferencia de datos MC14411 previamente instalado. Los circuitos integrados CI 2A e CI 2B funcionan como una cadena de divisor programable. Con un nivel lógico 1 TTL en el CI 2 de entrada divide los 4800 Hz por 2, resultando una salida de 2400 Hz. Cuando el nivel de entrada se cambia a un nivel lógico «0», lo divide por 4, produciendo una salida de 1200 Hz. Las frecuencias de FSK son generadas a una velocidad de salida en serie de 300 bps y están conectadas directamente a la grabadora a través de la entrada auxiliar o de micrófono. (Estas frecuencias y la velocidad de transferencia de datos son, a menudo, referidas como Kansas City Standard.)

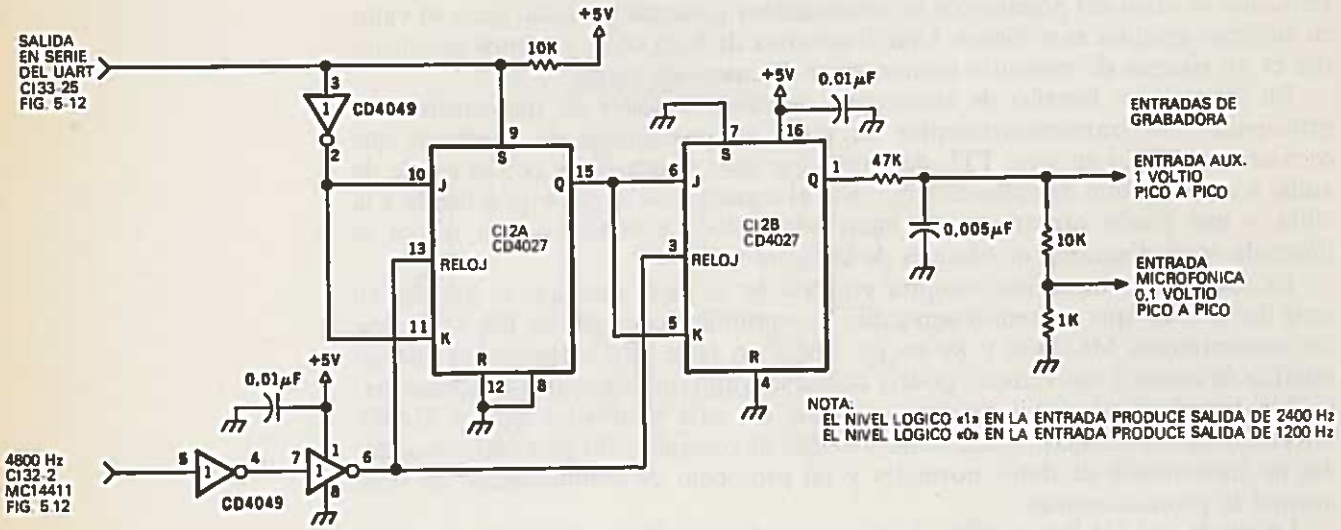


Figura 5.16 *Un controlador de salida en serie de 300 bps para una grabadora de audio.*

El obtener los sonidos grabados a partir de la cinta de audio requiere el circuito mostrado en la figura 5.17. En general, consiste en un par de filtros pasabanda y un comparador de tensiones. La grabadora se ajusta a un nivel de salida de aproximadamente 1 V de pico a pico. Este nivel no es crítico porque es amplificado y limitado cuando pasa a través del CI 1, CI 2 e CI 3 son filtros pasabanda con frecuencias centrales de 2400 y de 1200 Hz, respectivamente. La salida del CI 1 es alimentada a ambos, pero debe pasar por uno sólo de ellos. CI 4 compara las salidas de los dos filtros y genera un nivel lógico «1» de TTL, cuando un tono de 2400 Hz es recibido, lo mismo que cuando se recibe uno de 1200 Hz: la sintonización del interfaz será explicada más adelante.

La elección de las frecuencias FSK y de la velocidad de transferencia de datos no se realiza al azar. Estas son una función de la velocidad de respuesta del receptor y del ancho de la banda de la grabadora. La mayoría de las grabadoras de casete tienen una frecuencia de respuesta de unos 8 kHz. Unidades menos costosas pueden ser de un nivel de 5 ó 6 kHz. Es imprudente tratar de grabar sonidos que pasen de este límite superior. El centro del margen de frecuencias ofrece más fiabilidad, por lo que el tono del nivel lógico «1» del FSK debe ser ajustado a menos de 3 kHz (2400 Hz). En nuestro caso, además, lleva cierto tiempo al receptor reconocer una frecuencia particular. El circuito de la figura 5.17 tarda 2 o 3 ciclos en responder. Ello quiere decir que, a la baja frecuencia de 1200 Hz, cada bit de nivel lógico «0» necesitará 3 ciclos a 1200 Hz para ser identificado.

Si consideramos una condición del caso más desfavorable de emisión de todos «ceros», la velocidad de transmisión podría tener que ser más lenta que 400 bps para ser exactamente recibida. La velocidad de transferencia de datos normal más próxima a este valor es de 300 bps. La elevación al tono de 1200 Hz para

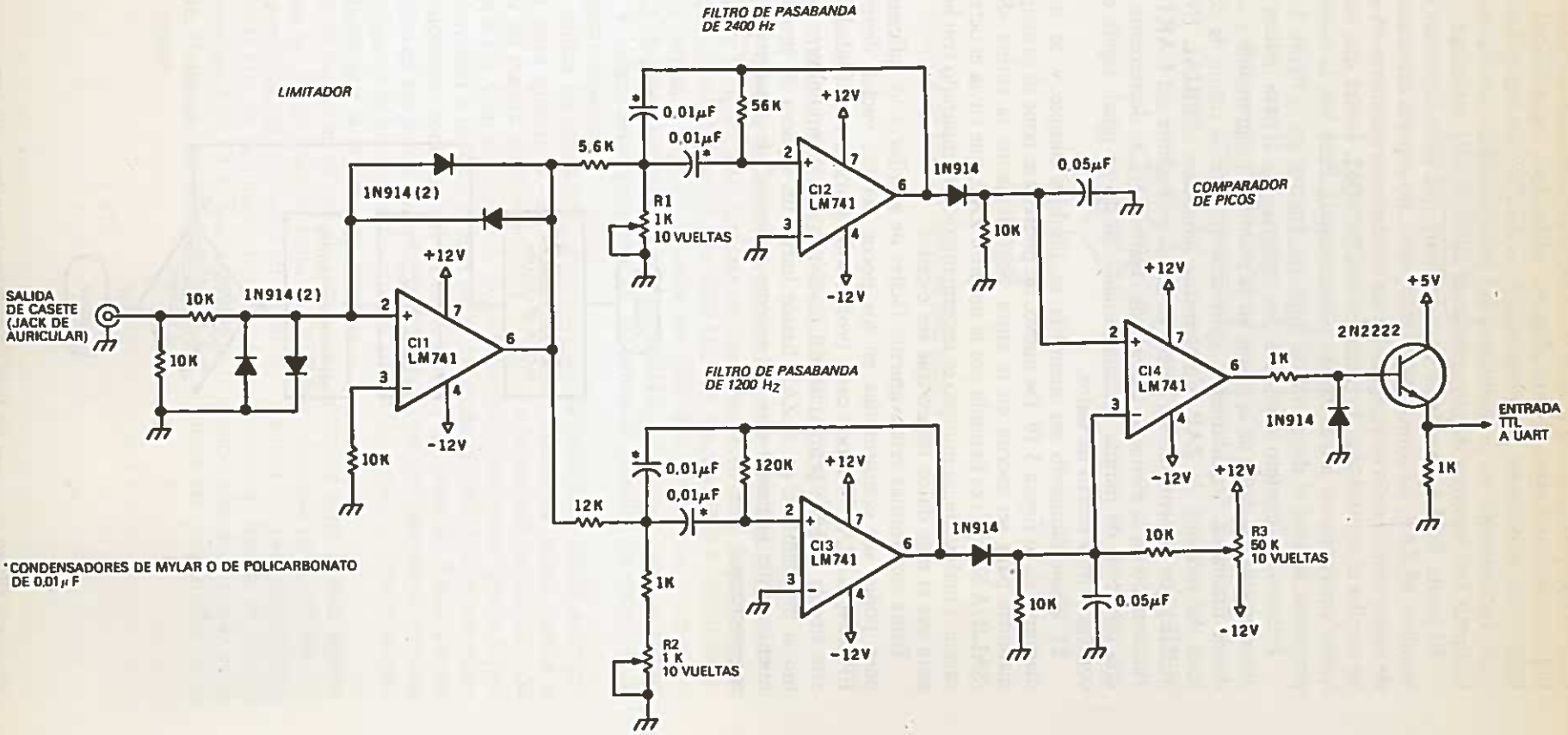


Figura 5.17 *Un receptor de entrada en serie de 300 bps de una casete de audio.*

incrementar la velocidad de transmisión solamente complica el diseño del filtro cuanto más próximo está a 2400 Hz. Este interfaz ha sido probado a 600 bps, pero requiere un ajuste especial para alcanzar velocidades más elevadas. Las frecuencias bajas y las velocidades de transferencia moderadas son escogidas específicamente para incrementar la probabilidad de construcciones satisfactorias en vez de competir con sistemas de almacenaje de datos de alta velocidad.

El punto final a considerar es el software que hace funcionar el hardware. El monitor del ZAP, tal como está en este punto, no soporta directamente un interfaz de casete aun cuando controle todas las operaciones auxiliares en serie. Hasta que no escriba el controlador de casete en una EPROM, tendrá que escribir en un pequeño programa de inicialización («bootstrap»). Para leer la casete, la lógica del programa seguirá el diagrama de flujo que aparece en la figura 5.18.

Primero, un puntero se pone en los registros H y L para indicar en dónde serán almacenados los datos de la casete en la memoria programable y una dirección en donde finalizará. Seguidamente, aprovechándose de la rutina de comunicación en serie del monitor de ZAP, simplemente llamamos «SERIAL IN» (ENTRADA SERIE) que retorna con un octeto de datos procedente del UART. Este octeto es almacenado en la memoria y el par de registros HL se decrementa y compara con una dirección de parada predeterminada. Si no es igual, repite el proceso para conseguir otro octeto de datos.

El almacenamiento en memoria es también simple y se representa en el diagrama de la figura 5.19. De nuevo, un puntero se pone al principio y la zona de memoria para ser escrita en la cinta. Seguidamente, la rutina «SERIAL OUT» (SALIDA SERIE) es llamada por el monitor ZAP, que envía el octeto de datos a la casete. Finalmente, el puntero es decrementado y comparado con la dirección final para ver si más datos tienen que ser escritos.

Estas son rutinas relativamente fáciles de escribir y lo suficientemente cortas para poder ser comprimidas en los pocos octetos vacíos dentro del monitor EPROM del ZAP. En todo caso, pronto constatará la versatilidad y capacidad que este simple interfaz proporciona a un sistema de computadora. Los 2 K de memoria programable en el ZAP básico harán un espacio de programa residente, mientras que la casete será un sistema potencial de almacenaje de ficheros de megaoctetos.

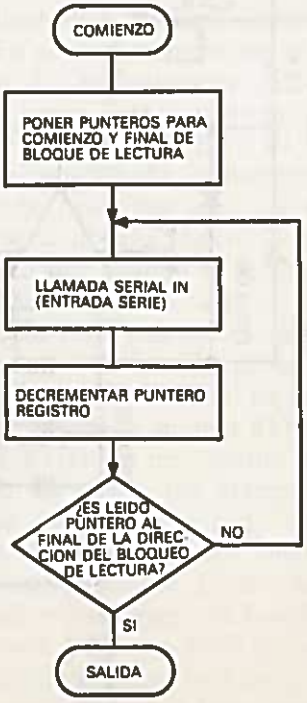


Figura 5.18 *Un diagrama de flujo de software para leer una casete.*

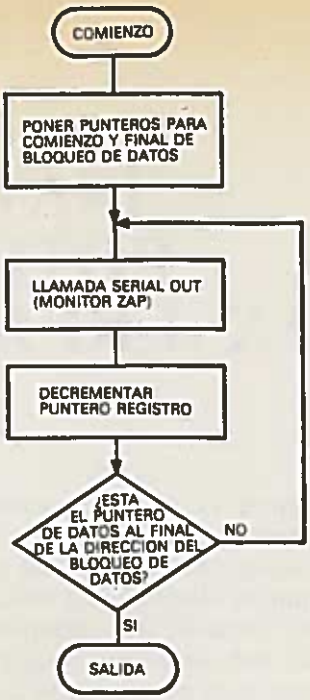


Figura 5.19 *Un diagrama de flujo de software para escribir en una casete.*

SINTONIZACION DEL INTERFAZ DE CASETE

Para probar el interfaz de casete, es necesario, en primer lugar, construir el circuito a partir de la figura 5.16. Utilizar un frecuencímetro para determinar que la entrada para el terminal 5 de CI 1 es 4800 Hz. Sin ningún UART instalado, la frecuencia en el terminal 1 de CI 2b debe ser 2400 Hz. La puesta a tierra del terminal 1 de CI 2b debe cambiar esta salida a 1200 Hz. En ambos casos, tensiones de 1 y de 0,1 V deben estar presentes en las entradas auxiliar de casete y de micrófono, respectivamente.

El receptor utiliza las frecuencias generadas por la sección de salida anteriormente descrita para ajustar la calibración. Con la sección de salida puesta a 2400 Hz, poner un puente desde el interfaz de salida a la entrada del circuito del receptor (figura 5.17). Con el empleo de un osciloscopio, comprobar que la forma de onda en el terminal 6 de CI 1 es una onda cuadrada de 2400 Hz. A continuación, con el osciloscopio conectado al terminal 6 de CI 2, ajustar R1 hasta que sea máxima la tensión en ese punto. Desplazando la sonda del osciloscopio al terminal 6 de CI 3 y cambiando la frecuencia de entrada a 1200 Hz, repetir el procedimiento ajustando R2 hasta que la tensión sea máxima.

R3 establece el punto en el que el comparador conmuta entre niveles lógicos cuando cambian las frecuencias de entrada. La forma adecuada para establecerlo es utilizar un generador de funciones en la entrada y ajusta R3 para conmutar a exactamente 1800 Hz. El resultado debe ser la conmutación limpia de niveles lógicos en el terminal 6 de CI 4, mientras que la frecuencia efectúa un ciclo entre 1200 y 2400 Hz. Hablando en términos generales, el ajuste del comparador no es especialmente crítico.

Capítulo 6

EL SOFTWARE ZAP MONITOR (MONITOR DEL ZAP)

La función de un sistema operativo es proporcionar al programador un conjunto de herramientas para ayudarle a desarrollar, depurar y ejecutar un programa. Por lo general, el sistema operativo ayuda al programador gestionando los recursos de la computadora y eliminando su implicación con manipulaciones repetitivas de código de máquina. Los sistemas operativos abarcan un amplio espectro de complejidad. Pequeños sistemas, por ejemplo, sólo proporcionan un medio rudimentario para un programador para introducir y leer datos de 8 bits desde la memoria; por el contrario, sistemas grandes pueden controlar dinámicamente la asignación de todos los periféricos y memoria.

Los sistemas grandes asignan recursos de computadora a más de un usuario en un ambiente de multiprogramación, multitareas o de tiempo compartido. Un sistema de esta magnitud supera, en mucho, las capacidades de la computadora descritas en este libro. Al ser así, ¿cuál sería un sistema operativo adecuado para la computadora ZAP? Como se indicó anteriormente, el objetivo de un sistema operativo es controlar los recursos de la computadora. La computadora ZAP descrita en los capítulos anteriores, y perfeccionada con los periféricos mínimos, contiene los recursos siguientes:

- Microprocesador Z80.
- 1024 octetos de memoria EPROM.
- 1024 octetos de memoria programable (2048 opcional).
- Teclado de 19 teclas.
- Visualizador de datos de dos caracteres.
- Visualizador de direcciones de cuatro caracteres.
- UART para E/S en serie.

El sistema operativo debe proporcionar acceso a estos recursos y dar al usuario una forma de controlarlos durante la ejecución de programas. El sistema operativo concebido para ZAP incluirá los medios y funciones siguientes:

1. Arranque en frío.
2. Arranque en caliente.
3. Visualizador de memoria y sustitución.
4. Visualizador de registro y sustitución.
5. Ejecución (comenzar la ejecución del programa en un punto designado).
6. Entrada y salida en serie.

Cada uno de los anteriores apartados se explicará con detalle en relación con sus funciones e implementación del programa.

I. FUNCIONES DEL SISTEMA OPERATIVO

Operación de arranque en frío

El sistema operativo debe estar disponible inmediatamente después de que se aplique la alimentación. En el pasado, algunos sistemas proporcionaban esta

capacidad almacenando, en memoria de sólo lectura, una pequeña rutina de autoinicialización «bootstrap». Esta rutina se utilizó entonces para cargar el sistema operativo en memoria desde otro dispositivo, tal como una lectora de cinta de papel o una grabadora de casete. La tecnología moderna elimina este paso tedioso. El sistema operativo para su computadora reside permanentemente en el chip de EPROM (ROM borrable y programable) y está preparado para ser ejecutado tan pronto como se aplique la alimentación y se pulse el botón «RESET». La acción de oprimir y de soltar el botón «RESET» pone a cero el PC (contador de programa) del Z80.

Con el siguiente ciclo de máquina, el procesador comienza la ejecución de la instrucción situada en 00_{16} (posición 00 hexadecimal) en memoria. El sistema operativo del microprocesador Z80 proporciona las instrucciones para comenzar la ejecución. Esta serie particular de instrucciones de programa constituye un procedimiento de «arranque en frío» y establece las condiciones requeridas de puesta a punto para el sistema operativo. Esto último inicializa el SP (puntero de pila) a una zona en memoria programable para mantener la pila «push-down/pop-up» (introducción por abajo/extracción por arriba). Esta pila se requiere para la ejecución de cualquiera de las instrucciones «RESTART» y «CALL», proporcionada por el conjunto de instrucciones del Z80. Si no se produjera la inicialización antes de la ejecución de una instrucción «CALL» o «RESTART», los efectos de la instrucción serían impredecibles. En este sistema operativo, el puntero de pila se pone a la posición de memoria programable $D7C4_{16}$.

Operación de arranque en caliente

Después de inicializar la dirección de SP, el sistema operativo introduce un modelo de reconocimiento de órdenes. Antes de examinar esta característica del sistema operativo, deben explicarse algunas de las otras características de reanudación funcional. El Z80 proporciona al usuario ocho instrucciones «RESTART» vectorizadas en dirección (ver Capítulo 3 para una descripción de las instrucciones). Por ejemplo, la ejecución de una RST 08_{16} almacenará el PC actual en la pila («STACK») y la ejecución del programa comenzará en la posición 08_{16}.

Las siguientes instrucciones «RESTART» están disponibles dentro del sistema operativo:

RST 10_{16}
RST 18_{16}
RST 20_{16}
RST 28_{16}
RST 30_{16}
RST 38_{16}

La ejecución de cualquiera de estas instrucciones hace que el sistema operativo salte a una posición en memoria programable. En esta posición, el usuario ejecuta una instrucción de salto para vectorizar la computadora a una nueva posición.

RST 00_{16} y RST 08_{16} han sido reservadas para utilizarlas por el sistema operativo en funciones especiales y no darán lugar a un salto a una posición en memoria programable. Estas dos instrucciones RST pueden ser utilizadas en la depuración de programas. RST 00_{16} realizará la misma función que si pulsamos el botón de «RESET», o reinicializará el puntero de pila e introducirá el módulo de reconocimiento de órdenes a través de la ejecución de la rutina del «arranque en frío».

La ejecución de una RST 08_{16} por el Z80 dará lugar a la introducción del módulo «de arranque en caliente». Este módulo conserva los datos existentes en todos los registros en el «área de conservación de registro» ubicada en la memoria programable (ver la lista del sistema operativo ZAP en el Apéndice D). El módulo también extraerá de la pila la dirección de relanzamiento del usuario y la guardará en el área de conservación de registro. El sistema operativo introduce entonces el módulo de reconocimiento de órdenes para esperar la próxima orden. El uso de este dispositivo permite al programador conservar el registro, el puntero, el

indicador y los datos del contador de programa antes de utilizar cualquier característica depuradora adicional en el sistema operativo. Una descripción detallada del módulo de «arranque en caliente» se proporciona en la sección II.2 de este capítulo.

Desarrollo del programa y servicios de depuración de errores

Los procedimientos de arranque en frío y en caliente dan salida a la secuencia de entrada de órdenes. Con estos procedimientos de órdenes, el programador es capaz de examinar y sustituir datos en la memoria, o en los registros, y de comenzar la ejecución del programa en la posición especificada por el usuario. A la entrada del módulo de introducción de órdenes, el sistema operativo muestra «FFFF» en la sección de direcciones y «FF» en la sección de datos de la visualización hexadecimal en LEDs de 6 caracteres. El usuario entonces implementa una de las tres funciones de órdenes manteniendo pulsada la tecla «SHIFT» y presionando las teclas «0», «1» ó «2». «SHIFT 0» (la tecla SHIFT y la tecla 0 pulsadas simultáneamente) comunican al sistema operativo que introduzca la función de visualización de la memoria y de sustitución; «SHIFT 1» introduce la función de visualización de registro y de sustitución y «SHIFT 2» introduce el módulo de «ir a ejecutar» («go execute») (ver más adelante).

Visualización de memoria y sustitución

Esta función permite al usuario examinar el contenido tanto de la memoria de sólo lectura como de la memoria programable. Durante la operación, la dirección y el contenido de esta posición se muestran en las respectivas visualizaciones.

La función de visualización de memoria y sustitución se introduce ejecutando «SHIFT 0», cuando el sistema está en el modo de reconocimiento de órdenes (visualización de dirección=FFFF y visualización de datos=FF). En este momento, el sistema operativo está esperando que el usuario introduzca una dirección de 1 a 4 dígitos hexadecimales desde el teclado. A medida que se introducen, estos dígitos se desplazan secuencialmente en la pantalla. Si se introduce más de cuatro dígitos, sólo el último valor de 4 dígitos (mostrado en la visualización de la dirección) será utilizado como la dirección. La introducción de datos de dirección se termina por la pulsación de la tecla «NEXT». Ello hace que el contenido de la dirección inidicada sea visualizada en la presentación hexadecimal de datos de 2 dígitos. Si el usuario desea presentar en la pantalla posiciones siguientes de la memoria, solamente necesita continuar pulsando la tecla NEXT. Con ello se llevará al programa de la visualización de la memoria a la siguiente posición más alta de la memoria y visualizará el nuevo contenido de dirección y de la memoria. Si el usuario desea cambiar el contenido de una posición de memoria mostrada en la pantalla, puede introducir nuevos datos escribiendo un valor de 2 dígitos para esa posición antes de pulsar la siguiente tecla. Este nuevo valor es cargado en la dirección indicada cuando la tecla «NEXT» es pulsada. Al pulsar la tecla «NEXT» continúa la presentación secuencial en la pantalla de direcciones y datos.

La terminación de esta función se realiza pulsando los botones «RESET» o «EXEC». El control se devuelve a la sección de reconocimiento de órdenes del sistema.

Ejemplo de memoria de visualización

Tecla	Visualización de dirección	Visualización de datos
	FFFF	FF
«SHIFT 0»	0000	FF
1	0001	FF
A	001A	FF
F	01AF	FF
«NEXT»	01AF	01
«NEXT»	01B0	1C
«RESET»	FFFF	FF

Ejemplo de sustitución de memoria

Tecla	Visualización de dirección	Visualización de datos
	FFFF	FF
«SHIFT 0»	0000	FF
4	0004	FF
0	0040	FF
0	0400	FF
«NEXT»	0400	01
2	0400	02
1	0400	21
«NEXT»	0401	05
6	0401	06
A	0401	6A
«EXEC»		

Los resultados serán:	Dirección	Datos
	0400	21
	0401	6A

Visualización de registro y sustitución

La función de visualización de registro y sustitución permite al usuario examinar y cambiar los contenidos de los registros conservados del Z80. Esto es logrado por la ejecución de RST1 (arranque en caliente) durante la ejecución del programa. Durante la ejecución de esta función, los contenidos de los registros son mostrados en el visualizador de dirección. Registros de 8 bits serán mostrados en los dos dígitos más bajos del visualizador de dirección. (Los dos dígitos más altos serán «ceros» durante la presentación de los registros de 8 bits.) Un código que indica qué registro está siendo visualizado se puede apreciar en la presentación visual de datos. La tabla 6.1 describe los códigos que han sido asignados a la función de visualización de registros y sustitución, así como la tecla que inicia una secuencia particular de la visualización de registros.

Código (mostrado en el visualizador de datos)	Registro del Z80 (mostrado en el visualizador de dirección)	Tecla de iniciación
02	IX	2
03	IY	3
04	SP	4
05	PC	5
06	I	6
07	R	7
08	L	8
09	H	9
0A	A	A
0B	B	B
0C	C	C
0D	D	D
0E	E	E
0F	F	F
40	L'	«SHIFT 0»
41	H'	«SHIFT 1»
42	A'	«SHIFT 2»
43	B'	«SHIFT 3»
44	C'	«SHIFT 4»
45	D'	«SHIFT 5»
46	E'	«SHIFT 6»
47	F'	«SHIFT 7»

Tabla 6.1 *Correspondencia de tecla de iniciación/registro del Z80/código de visualización.*

La función de visualización de registros y sustitución se introduce pulsando «SHIFT 1» cuando el sistema está en el modo de reconocimiento de órdenes (visualización de dirección=FFFF y visualización de datos=FF). En este momento el sistema operativo está esperando a que el programador introduzca el código de registro de un solo dígito (ver tabla 6.1). Si se introduce más de un dígito, solamente el último código indicado en la visualización de datos será utilizado como el identificador de registro. Cuando el procesador central detecta que se ha pulsado la tecla «NEXT», el contenido del registro indicado se muestra en la visualización de dirección.

Si el usuario desea mostrar en la pantalla los subsiguientes registros, sólo necesita pulsar la tecla «NEXT». Esto hace que surja el siguiente registro con su código de registro y su contenido. Para cambiar el contenido de un registro visualizado, el valor se introduce y carga cuando se pulsa la tecla «NEXT». Para registros de 16 bits, los últimos 4 dígitos hexadecimales serán aceptados si más de 4 caracteres se han introducido. Cuando se sustituyen los datos del registro, la tecla «NEXT» hace también que el código de registro sea indexado al siguiente registro (ver tabla 6.1) y sea visualizado su contenido.

El usuario puede hacer que finalice esta función pulsando la tecla «EXEC». El control se devuelve a la parte de reconocimiento de órdenes del sistema operativo.

Ejemplo de registro de visualización

Tecla	Visualización de datos (Código de registro)	Visualización de direcciones (Contenido de registro)
	FF	FFFF
«SHIFT 1»	00	FFFF
A	0A	FFFF
«NEXT»	0A	005C
«NEXT»	0B	0063
«RESET»	FF	FFFF

Ejemplo de sustitución de registro

Tecla	Visualización de datos (Código de registro)	Visualización de direcciones (Contenido de registro)
	FF	FFFF
«SHIFT 1»	00	FFFF
5	05	FFFF
«NEXT»	05	043A
4	05	0004
2	05	0042
C	05	042C
«NEXT»	06	00FF
«NEXT»	07	0003
«EXEC»		

Función «go execute» —Ir a ejecutar— («EXEC»)

La función «go execute» —Ir a ejecutar— («EXEC») permite al usuario cambiar el contenido del registro del PC (contador de programa) para poder realizar la ejecución directa de instrucciones en la dirección seleccionada por el usuario.

Esta función «EXEC» se introduce pulsando «SHIFT 2» cuando el sistema está en el modo de reconocimiento de órdenes. Ahora, el usuario debe introducir una dirección de uno a cuatro dígitos hexadecimales. Si se introducen más de cuatro

dígitos, sólo el valor mostrado en la visualización de dirección se utiliza como la dirección para comenzar la ejecución de programa. La ejecución comienza cuando se pulsan las teclas «NEXT» o «EXEC». Con ello los registros del Z80 se almacenan en la zona de conservación de registros (ver el listado del sistema operativo en el Apéndice D) y la ejecución comienza en la dirección especificada por el usuario.

Ejemplo de GO EXECUTE

Tecla	Visualizador de dirección	Visualizador de datos
	FFFF	FF
«SHIFT 2»	0000	FF
1	0001	FF
A	001A	FF
C	01AC	FF
F	1ACF	FF
«NEXT» o «EXEC»		

Servicio de E/S en serie

La computadora ZAP posee una capacidad de entrada y salida en serie que es realizada gracias al dispositivo UART. Este interfaz permite la comunicación en serie entre la computadora y dispositivos periféricos, tales como una impresora o un TRC. Para auxiliar al usuario en la utilización de estos recursos, el sistema operativo tiene un módulo de diagnóstico de UART, un módulo de entrada en serie y un módulo de salida en serie. Los módulos de entrada y de salida son establecidos como subrutinas que pueden ser solicitadas durante la ejecución del programa y que están necesariamente limitados al teclado y a la presentación visual.

Módulo de diagnóstico del UART

El módulo de diagnóstico del UART proporciona los medios para la revisión del funcionamiento del UART. Para utilizar esta característica operativa, el usuario debe primero unir las líneas de entrada y salida en serie, de tal forma que las salidas de datos del UART puedan ser leídas por un mismo dispositivo. La subrutina del diagnóstico en serie se inicia usando la función «ir a ejecutar». La ejecución comienza en $032D_{16}$.

Una vez comenzado, el módulo de diagnóstico (UATST) comienza a enviar datos al UART y espera que haya datos disponibles. El estado del UART se comprueba para verificar que no existen condiciones de anomalía funcional. En el caso de que sea detectado un error, el estado del UART es mostrado en la pantalla en los dos dígitos de orden bajo de la visualización de dirección. (Ver tabla 6.2 para códigos de error.) Si no hay errores, los datos son leídos y visualizados en la presentación de datos de dos dígitos. Se hace una comparación entre los datos de entrada y de salida. Si los dos octetos son iguales, el carácter de salida es incrementado y otro octeto es enviado al UART para continuar la secuencia. Este procedimiento continúa hasta que se pulse el botón «RESET», o hasta que sea detectado un error. En el caso de que el carácter de entrada no sea igual al carácter de salida, $0F_{16}$ se visualiza en los dos dígitos más bajos de la visualización de dirección y se interrumpe el diagnóstico. La figura 6.1 detalla el flujo lógico de esta rutina de software.

Código visualizado	Error
12_{16} ó 13_{16}	Error de paridad
$0A_{16}$ ó $0B_{16}$	Error de encuadramiento
06_{16} ó 07_{16}	Error de sobre-ejecución
00	Memoria intermedia de transmisor no vacía
$0F_{16}$	Carácter de entrada distinto a carácter de salida

Tabla 6.2 *Códigos de error del UART.*

Módulo de entrada en serie

El módulo de entrada en serie ha sido incluido de tal forma que el usuario pueda leer datos en serie de dispositivos externos. Para utilizar este recurso, el usuario debe poner aparte una memoria intermedia de memoria programable, en donde los datos de entrada hayan de almacenarse, e indicar el número de caracteres de entrada previstos. La dirección de memorias intermedios de entrada es almacenada en la dirección de $07F9_{16}$ en la memoria (ver el Apéndice D) y el número de caracteres es almacenado en la dirección $07FD_{16}$. La recepción de la comunicación comienza cuando el módulo TTYINP es solicitado.

Ejemplo de iniciación de entrada en serie

```
TTYINP  EQU  035F16      Dirección de módulo de entrada
BUFFER  EQU  07F916      Dirección de entrada de memoria intermedia
NCHAR   EQU  80          Número de caracteres a recibir
TTYIBU  EQU  07F916      Constante de dirección del sistema operativo
TTYIC   EQU  07FD16      Constante de dirección del sistema operativo
        LD   HL, BUFFER  Poner memoria intermedia para sistema ope-
                         rativo
        LD   (TTYIBU), HL Poner la cuenta de caracteres para el sistema
        LD   A, NCHAR    operativo
        LD   (TTYIC), A  Llamada de rutina de entrada en serie del
        CALL TTYINP      UART
```

Los datos leídos por el módulo de entrada en serie serán almacenados en la memoria intermedia especificado por el usuario hasta que la secuencia de entrada esté terminada. Cuando esto sucede, el control se devuelve al programa del usuario en la siguiente instrucción. La finalización del proceso de entrada puede ser debida a alguna de las condiciones siguientes:

- Un error de estado es detectado.
- El número de caracteres leídos es igual al contaje preestablecido.
- La recepción del retorno del carro como un carácter de entrada (ASCII $0D_{16}$).

En el caso de que se detecte un error de estado, el registro A será igual a 80_{16} cuando el control se devuelve al usuario. Si la finalización resulta del llamado correcto de la memoria intermedia de caracteres, el registro A será igual a 00_{16}. Sin embargo, si la finalización es el resultado de un retorno del carro, el registro A será igual al número de caracteres que quedan por introducir. La figura 6.2 detalla flujo lógico del módulo TTYINP de software.

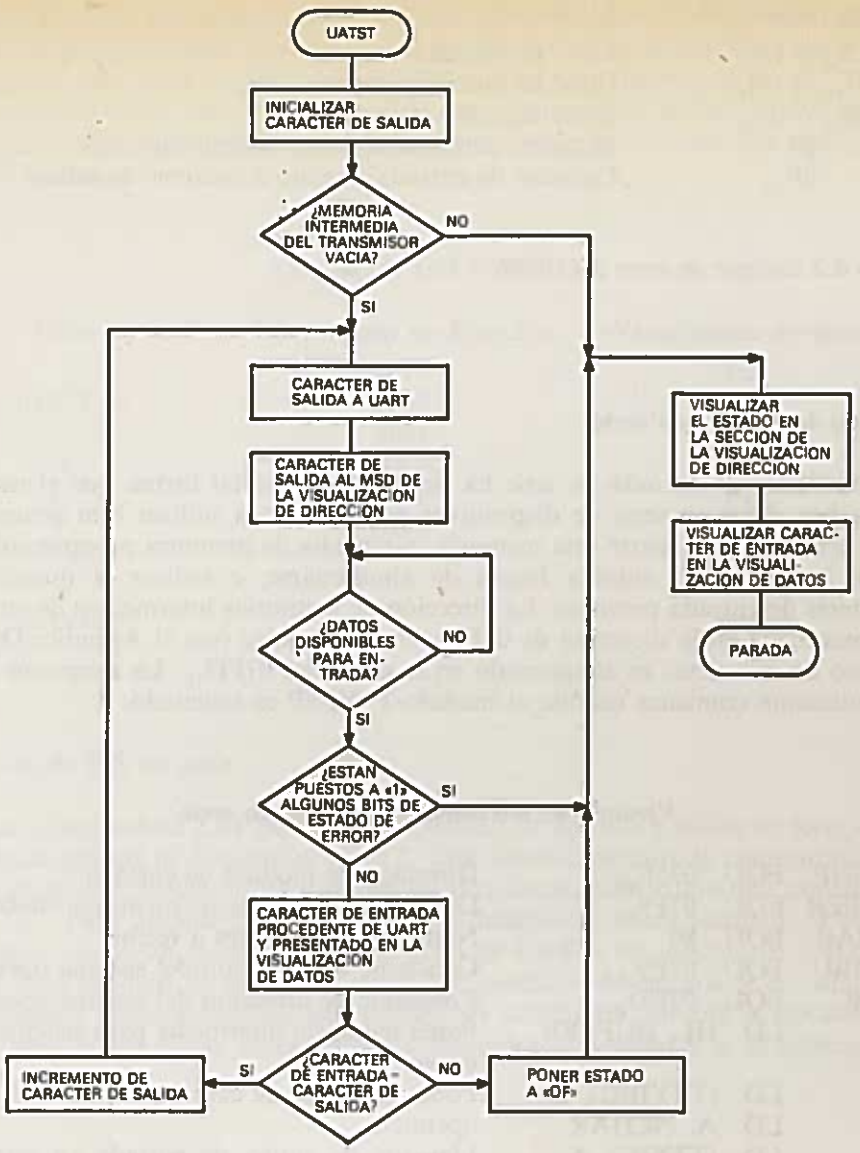


Figura 6.1 *Un diagrama de flujo del módulo de diagnóstico UART (UATST).*

Módulo de salida en serie

El módulo de salida en serie ha sido previsto para asistir al usuario en la comunicación de datos de salida en serie a dispositivos externos. Para utilizar este módulo, el operador indica una dirección de memoria intermedia de datos de salida y el número de caracteres (octeto) a transmitir. La dirección de memoria intermedia de salida debe almacenarse en $07FB_{16}$ en la memoria (ver el Apéndice D) y el número de caracteres a enviar es almacenado en la dirección $07FE_{16}$. La transmisión de datos comienza cuando TTYOUT es solicitada.

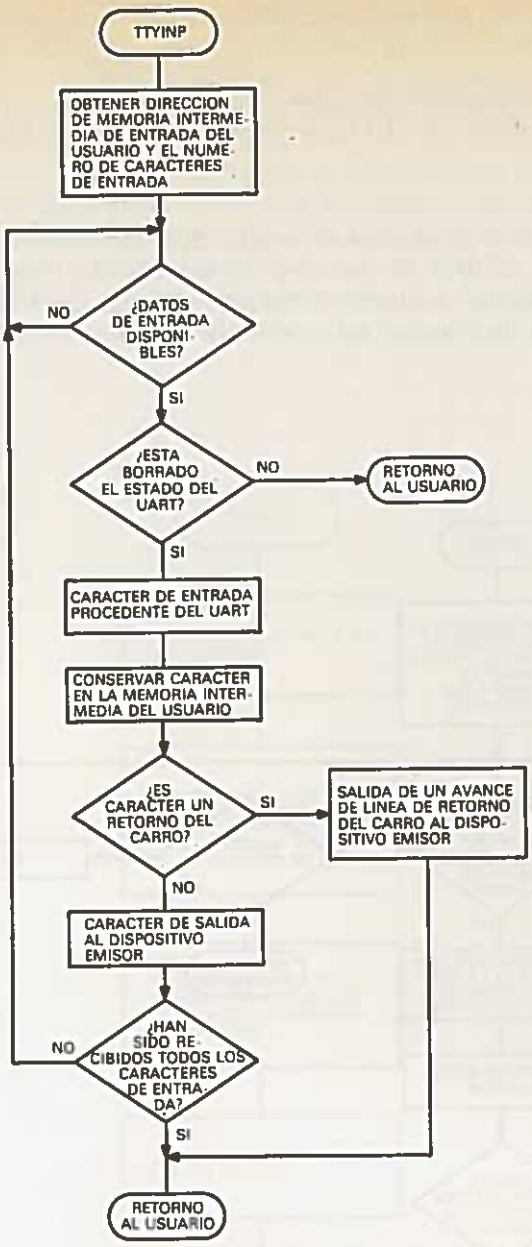


Figura 6.2 *Un diagrama de flujo del módulo de entrada en serie (TTYINP).*

Ejemplo de iniciación de salida en serie

```
TTYOUT  EQU  039E16      Dirección de módulo de salida
BUFFER  EQU  07FB16      Dirección de memoria intermedia de salida
NCHAR   EQU  35          Número de caracteres a transmitir
TTYOBF  EQU  07FB16      Constante de dirección del sistema operativo
TTYOCE  EQU  07FE16      Constante de dirección del sistema operativo
        LD   HL, BUFFER  Establecimiento de dirección de memoria in-
                         termedia para sistema operativo
        LD   (TTYOEF), HL Poner la cuenta de caracteres para el sistema
        LD   A, ANCHAR   operativo
        CALL TTYOUT      Llamar la rutina de salida en serie del UART
```

El control se devolverá al usuario cuando:

- La memoria intermedia de salida esté vacía.
- La memoria intermedia del transmisor no se encuentra disponible, indicando un error.

En el caso de que se produzca una terminación normal, el registro A será igual a 00_{16}, cuando el control se devuelve al usuario. Sin embargo, si hay una terminación prematura y el retorno es requerido, el registro A será igual a 01_{16}. La figura 6.3 detalla el flujo lógico del módulo de salida en serie de software.

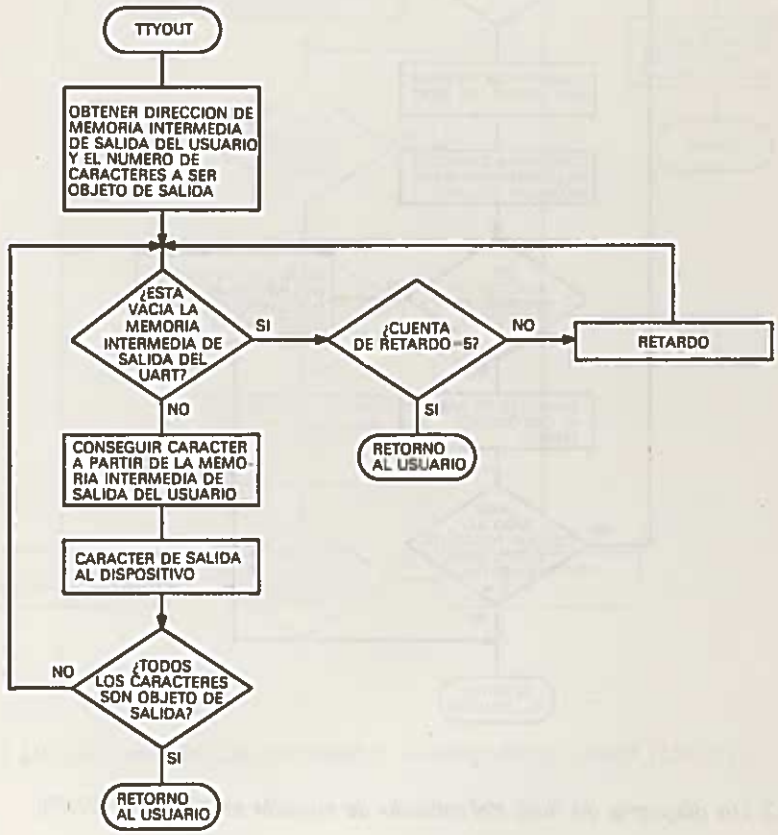


Figura 6.3 *Un diagrama de flujo de un módulo de salida en serie (TTYOUT).*

II. Descripción del módulo del sistema operativo

II.1. Módulo de arranque en caliente

El módulo de arranque en caliente (WARM1) es responsable de conservar todos los registros del Z80 en la zona de conservación de registros asignada en una

sección reservada de la memoria programable (ver Apéndice D). A la entrada, los registros del usuario A, H y L están conservados para proporcionar registros de trabajo para el resto de la operación del módulo. Seguidamente, el PC del usuario es extraído de la pila y conservado en las posiciones de memoria reservadas para el mismo. El par de registros AF se lleva a la pila y se saca hacia el par de registros HL. Este procedimiento permite que el registro de indicadores sea conservado en el área de conservación de registros. El resto de los registros de trabajo del usuario y alternativos son examinados y transferidos a la zona de conservación de registros. Una vez completada esta tarea, el módulo sale al módulo de reconocimiento de órdenes. (Ver Apéndice D para detalles adicionales.) La figura 6.4 detalla el flujo lógico del módulo de arranque en caliente.

Figura 6.4 *Un diagrama de flujo de un módulo de arranque en caliente (WARM1).*

II.2. Módulo de reconocimiento de órdenes

El módulo de reconocimiento de órdenes (WARM2) se introduce después de la terminación de la secuencia del arranque en frío o en caliente. Cuando se inicia, el módulo borra la memoria intermedia de entrada del teclado y los indicadores del mismo. Con ello se elimina la ambigüedad para operaciones futuras. El módulo pondrá la visualización de datos en FF y la visualización de dirección en FFFF. Una vez completado, el módulo introduce la subrutina KEYIM para conseguir un carácter de entrada del teclado. Cualquier carácter de entrada es revisado para ver si corresponde a una de las tres funciones permitidas. Si fuera así, el control se transfiere a la función adecuada; de cualquier otro modo, la entrada es ignorada y el módulo espera a la siguiente entrada procedente del teclado. (Ver el Apéndice D para detalles adicionales.) La figura 6.5 ilustra el flujo lógico del módulo de reconocimiento de órdenes.

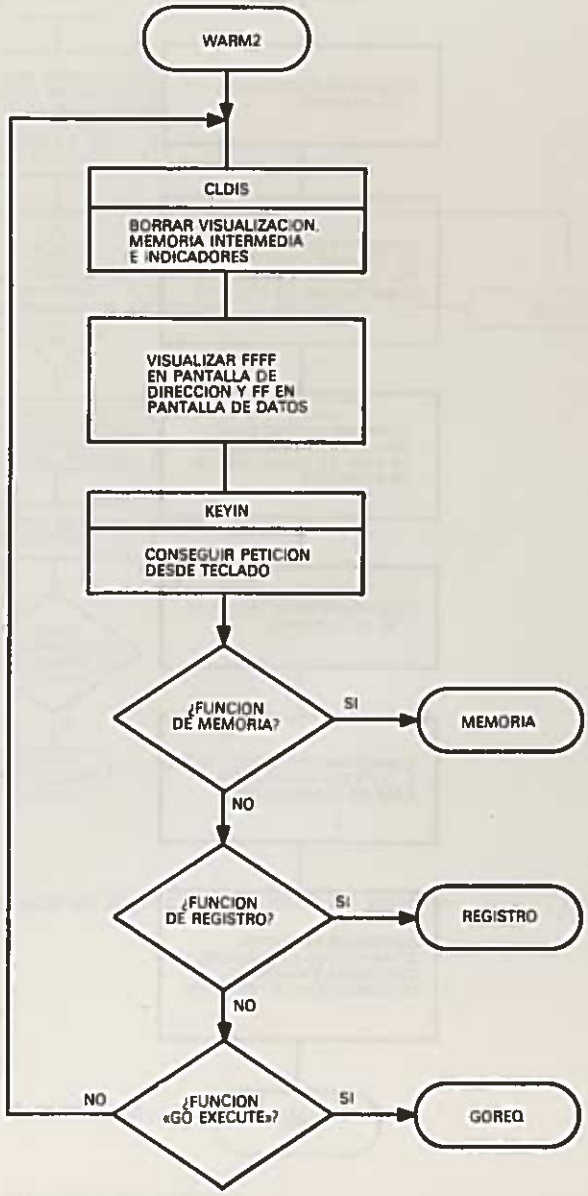


Figura 6.5 *Un diagrama de flujo del módulo de reconocimiento de órdenes (WARM2).*

II.3. Módulo de relanzamiento (RESTART)

Este módulo toma los valores almacenados en la zona de conservación de registros de la memoria programable. Luego restaura los registros del usuario de 8 y 16 bits antes de volver el control a la posición especificada en la zona de conservación del PC. Este procedimiento restaura los registros alternativos y luego los registros de trabajo. En uno u otro caso, los registros de indicadores son restaurados introduciendo los datos en la pila y luego sacándolos hacia el registro F. Con el fin de dar salida a la dirección de relanzamiento del usuario, el PC conservado se introduce en la pila y se ejecuta una «RET» (instrucción de retorno). (Ver el Apéndice D para detalles adicionales.) La figura 6.6 detalla un flujo lógico del módulo de retorno al comienzo (relanzamiento).

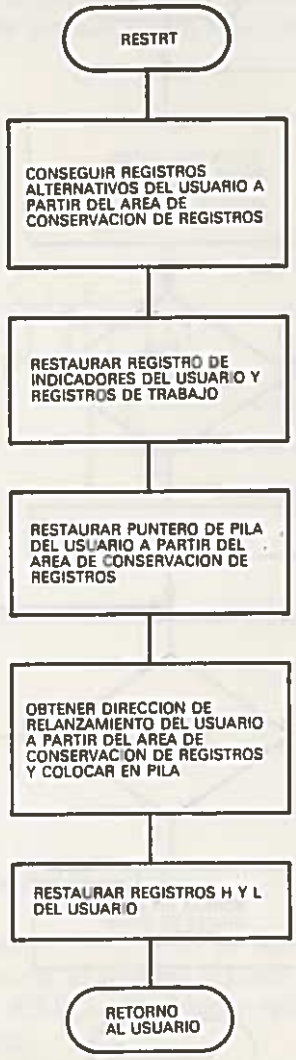


Figura 6.6 *Un diagrama de flujo del módulo de relanzamiento (RESTRT).*

II.4. Módulo de entrada del teclado

El módulo de entrada del teclado (KEYIN) proporciona el interfaz primario entre la computadora y el usuario. A la entrada, comienza a leer datos del port de entrada del teclado. Se mantiene en un bucle, comprobando el MSB (bit más significativo) de los datos. El MSB es el «strobe» de pulsación de tecla. Cuando pasa a un nivel lógico «1», los siete LSB (bits menos significativos) del port de entrada del teclado son retenidos como el carácter de entrada deseado. El módulo retorna luego al programa del usuario con el carácter del teclado en el acumulador. (Ver Apéndice D para detalles adicionales.) La figura 6.7 detalla el flujo lógico del módulo de entrada de teclado.

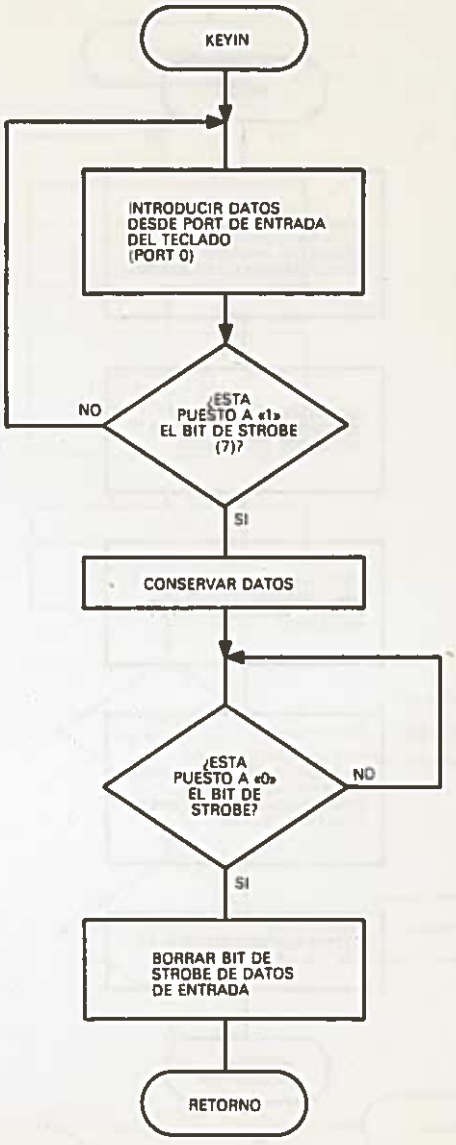


Figura 6.7 *Un diagrama de flujo de un módulo de entrada de teclado (KEYIN).*

II.5. Módulo de entrada de un solo carácter

La función de este módulo (ONECAR) es la de introducir uno o más caracteres desde el teclado. Este módulo también indica el último carácter y si iba acompañado por una tecla «NEXT» o «EXEC».

A la entrada, la memoria intermedia de entrada y los indicadores del teclado son objeto de borrado. (La visualización de datos puede, o no, ser borrada dependiendo de los requerimientos del módulo de llamada.) El módulo espera que un carácter de entrada le atraviese. Cuando recibe un carácter, le comprueba para ver si es un «NEXT», un «EXEC» o datos válidos. En el caso de que la entrada sea un «NEXT» o un «EXEC», el indicador del teclado adecuado se activa junto con el indicador de no datos y el control se devuelve al usuario (ver figura 6.8).

Si un carácter de datos no válidos es recibido, el módulo se reinicia. Una vez recibidos datos válidos, los datos se almacenan en una memoria intermedia de entrada de un octeto y el módulo espera al siguiente carácter de entrada. Este carácter es procesado de una manera similar a la recientemente descrita con la siguiente excepción: en el caso de que el carácter de entrada es un «NEXT» o un «EXEC», solamente el indicador apropiado es activado antes de volver el control al usuario. (Ver el Apéndice D para detalles adicionales.) La figura 6.9 muestra un flujo lógico de un módulo de entrada de un solo carácter.

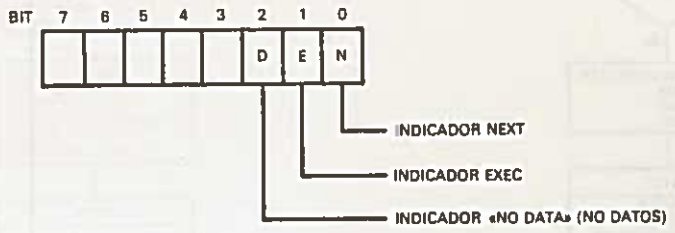


Figura 6.8 *La configuración de los indicadores del teclado.*

II.6. Módulo de entrada de dos caracteres

La función de este módulo (TWOCAR) es introducir uno o más caracteres mediante el teclado y transferir al usuario los dos últimos caracteres cuando se pulse una tecla «NEXT» o «EXEC». El módulo también indica al usuario el tipo de finalización que tuvo lugar.

A la entrada, la memoria intermedia de entrada y los indicadores del teclado son objeto de borrado. (La visualización de datos puede, o no, ser borrada dependiendo de los requerimientos del módulo de llamada.) Este módulo llama al módulo de entrada del teclado para obtener sus datos de entrada. El primer carácter es comprobado para determinar si es un «NEXT» o un «EXEC»; el indicador del teclado apropiado es puesto a «1» junto con el indicador de no datos y el control se vuelve al usuario (ver figura 6.8). Si un carácter no válido es recibido, el módulo se reinicia.

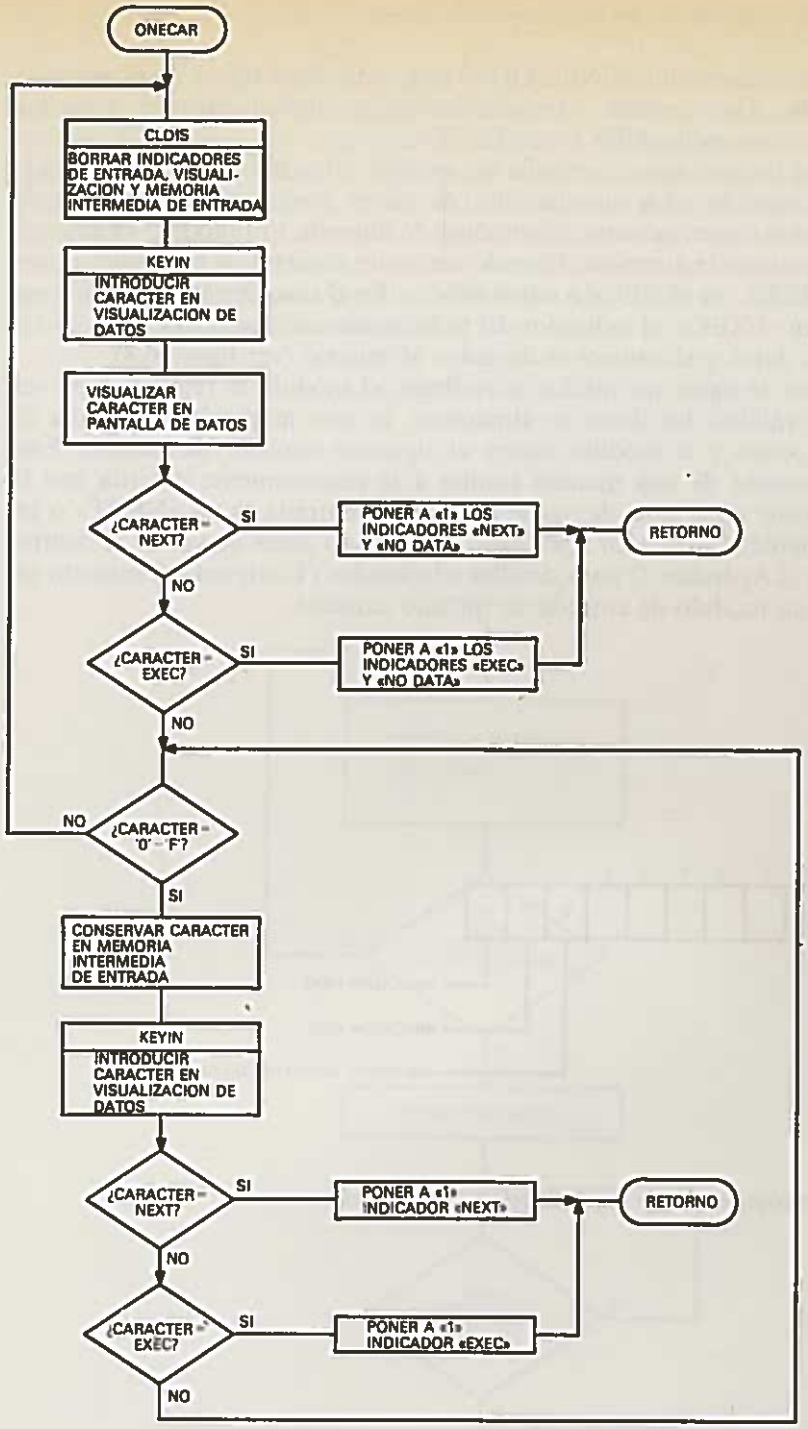


Figura 6.9 *Un diagrama de flujo de un módulo de entrada de un solo carácter (ONECAR).*

La recepción de datos válidos hará que el módulo proporcione un formato de los datos como un valor de dos dígitos en la memoria intermedia de entrada del teclado. Luego retorna al usuario con los indicadores apropiados puestos a «1». (Ver el Apéndice D para detalles adicionales.) La figura 6.10 detalla el flujo lógico de un módulo de entrada de dos caracteres.

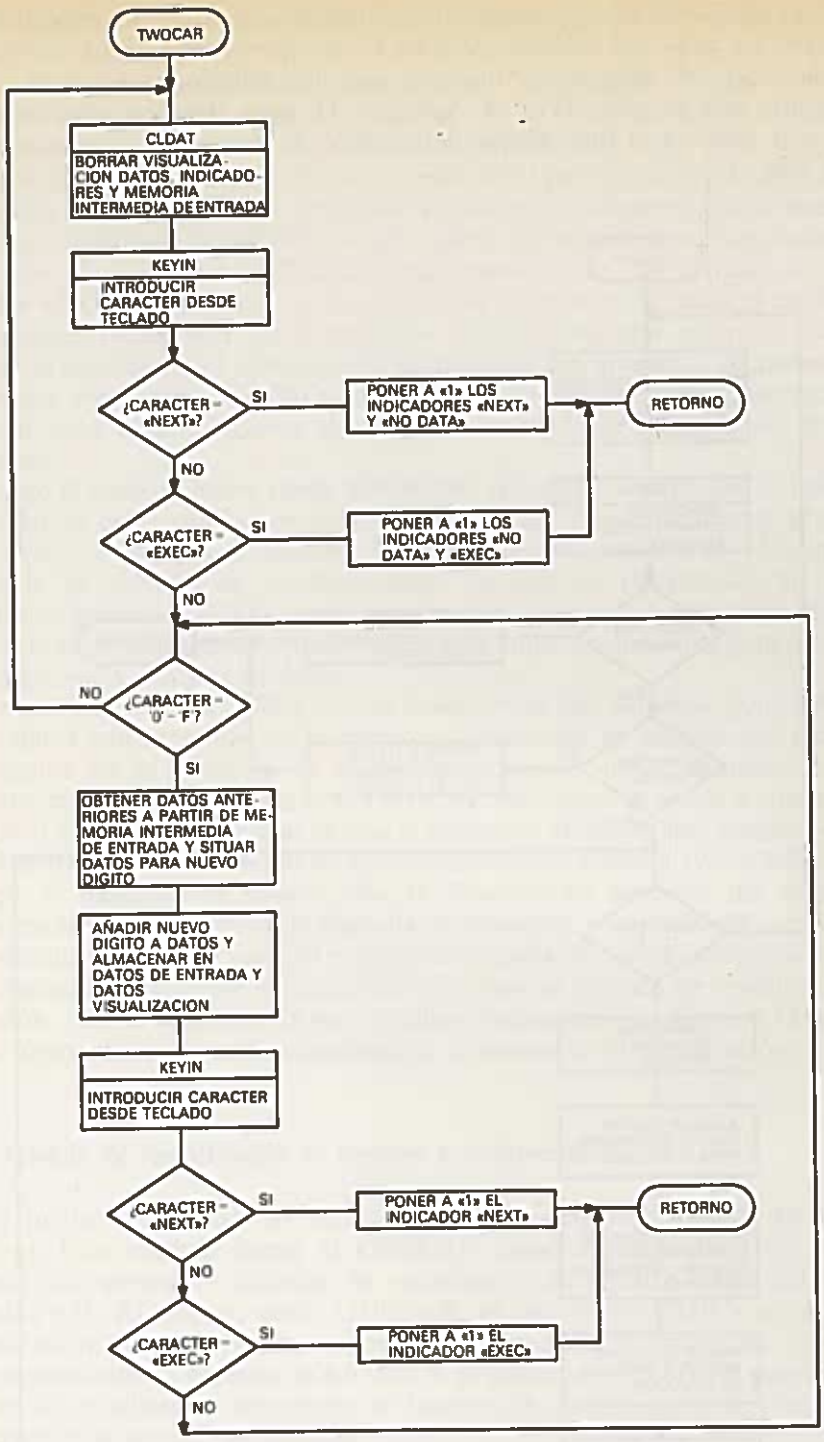


Figura 6.10 *Un diagrama de flujo del módulo de entrada de dos caracteres (TWOCAR).*

II.7. Modelo de entrada de cuatro caracteres

La función de este módulo (FORCAR) es introducir uno o más caracteres desde el teclado y transferir al usuario los últimos cuatro caracteres cuando se pulse una tecla «NEXT» o «EXEC». En el caso de que menos de cuatro caracteres sean introducidos, los dígitos de orden más alto serán puestos a 0. El módulo proporciona también notificación al usuario a través de los indicadores del teclado (ver figura 6.8).

La operación de este módulo es muy similar al módulo de entrada de dos caracteres. La principal diferencia estriba en la manera en que los nuevos datos (entrada desde el teclado) se fusionan con los anteriores datos de entrada procedentes del teclado. (Ver el Apéndice D para detalles adicionales.) La figura 6.11 muestra el flujo lógico del módulo de entrada de cuatro caracteres (FORCAR).

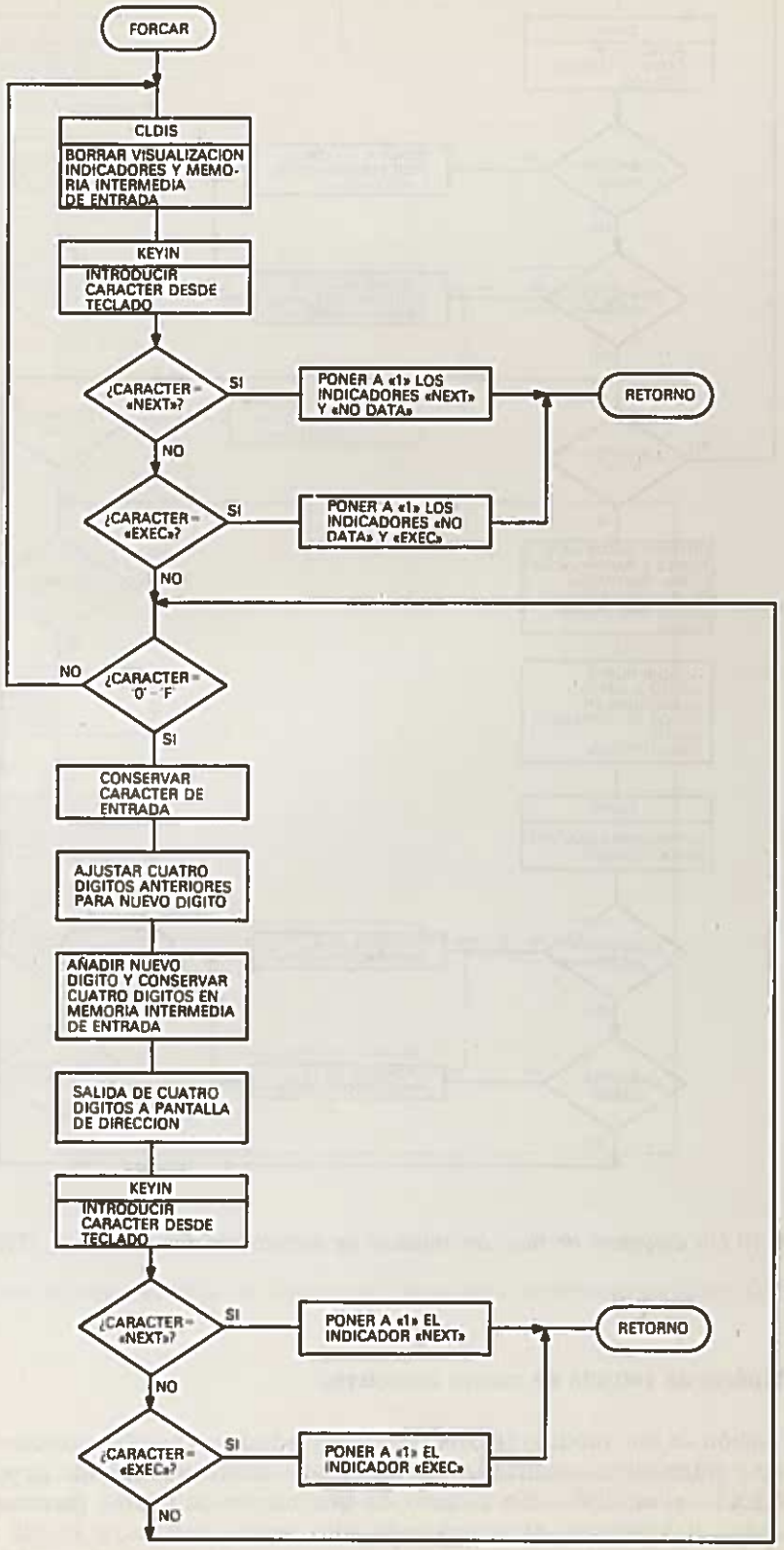


Figura 6.11 *Diagrama de flujo de un módulo de entrada de cuatro caracteres (FORCAR).*

II.8. Módulo de visualización de memoria y sustitución

La función de visualización de memoria y sustitución es uno de los tres principales módulos del sistema operativo. A la entrada, ver el módulo de reconocimiento de órdenes. Este método (MEMORY) hace una llamada a FORCAR (módulo de entrada de cuatro caracteres) para conseguir la dirección de memoria de base, en la que se comienza a visualizar el contenido de la memoria. Cuando retorna desde FORCAR, los indicadores del teclado son examinados para determinar si el indicador «EXEC» está puesto a «1». En el caso de que el indicador «EXEC» esté puesto a «1», el control es transferido hacia el módulo de relanzamiento (RESTRT). Si el indicador «EXEC» no está activado (=0), la posición de dirección y el contenido de la memoria son sacados a las presentaciones visuales apropiadas. El TWOCAR (módulo de entrada de dos caracteres) es solicitado para obtener nuevos datos procedentes de la posición de memoria visualizada.

Cuando el control vuelve desde TWOCAR, el módulo comprueba el indicador de «no datos» en la palabra de indicador de teclado. Si este indicador se pone a «1», el «EXEC» es objeto de examen. Si este último está puesto a «1», el control es transferido al módulo de reconocimiento de órdenes (WARM2). Si, por el contrario, el indicador «EXEC» está puesto a «0» (reset), la dirección de memoria del usuario es incrementada, presentada en la pantalla de dirección, y su contenido se visualiza en la pantalla de datos.

Si, en el retorno desde TWOCAR, el indicador de «no datos» se pone a «0», los nuevos datos son extraídos de la memoria intermedia de entrada del teclado y almacenados en la posición de memoria mostrada en la pantalla. En este momento, el módulo determina si el TWOCAR fue objeto de salida a través de un «EXEC» o «NEXT». En el caso de que el indicador «EXEC» está puesto a «1», el control es tranferido al módulo de reconocimiento de órdenes (WARM2). Si, sin embargo, el indicador se pone a «0», la dirección de memoria del usuario es incrementada y visualizada en la pantalla de dirección, y su contenido se presenta en la pantalla de datos. Luego, el módulo de entrada de dos caracteres es llamado para conseguir la siguiente pseudoinstrucción para el módulo de visualización de sustitución. (Ver el Apéndice D para detalles adicionales.) La figura 6.12 muestra el flujo lógico del módulo de visualización de memoria y de sustitución.

II.9. Módulo de visualización de registro y sustitución

Este módulo (REGIST) es uno de los tres principales módulos del sistema operativo. Este módulo llama al ONECAR (módulo de entrada de un solo carácter) para conseguir el código de visualización de registro inicial del usuario (ver tabla 6.1). Al retorno desde ONECAR, el indicador «EXEC» es objeto de comprobación. Si este indicador se pone a «1», el control es transferido al módulo de reconocimiento de órdenes (WARM2). Si el indicador «EXEC» es puesto a «0», el índice de visualización de registro de base es calculado a partir de código de la visualización de registro del usuario.

En este momento, el índice de registro es comprobado para ver si el registro requerido es de 8 o de 16 bits. Si el usuario requiere un registro de 16 bits, el código de registro apropiado es mostrado en el visualizador de datos y los datos de registro requeridos son obtenidos del área de conservación de registro y mostrados en el visualizador de dirección. El módulo entonces hace una llamada al FORCAR (módulo de entrada de cuatro caracteres) para conseguir nuevos datos para el registro. Al retorno, el indicador de «no datos» es revisado. Si este indicador está puesto a «1» y también lo está el «EXEC», el control es transferido al RESTRT (módulo de relanzamiento). Si los indicadores de «no datos» y «NEXT» están puestos a «1», el índice de visualización de registro es incrementado y mostrado en el visualizador de datos. Los nuevos datos de registros son obtenidos a partir del área de conservación de registro y son mostrados en el visualizador de dirección.

Si un registro de 8 bits ha sido solicitado, el código de registro (ver tabla 6.1) es mostrado en el visualizador de datos y los datos apropiados son obtenidos a partir

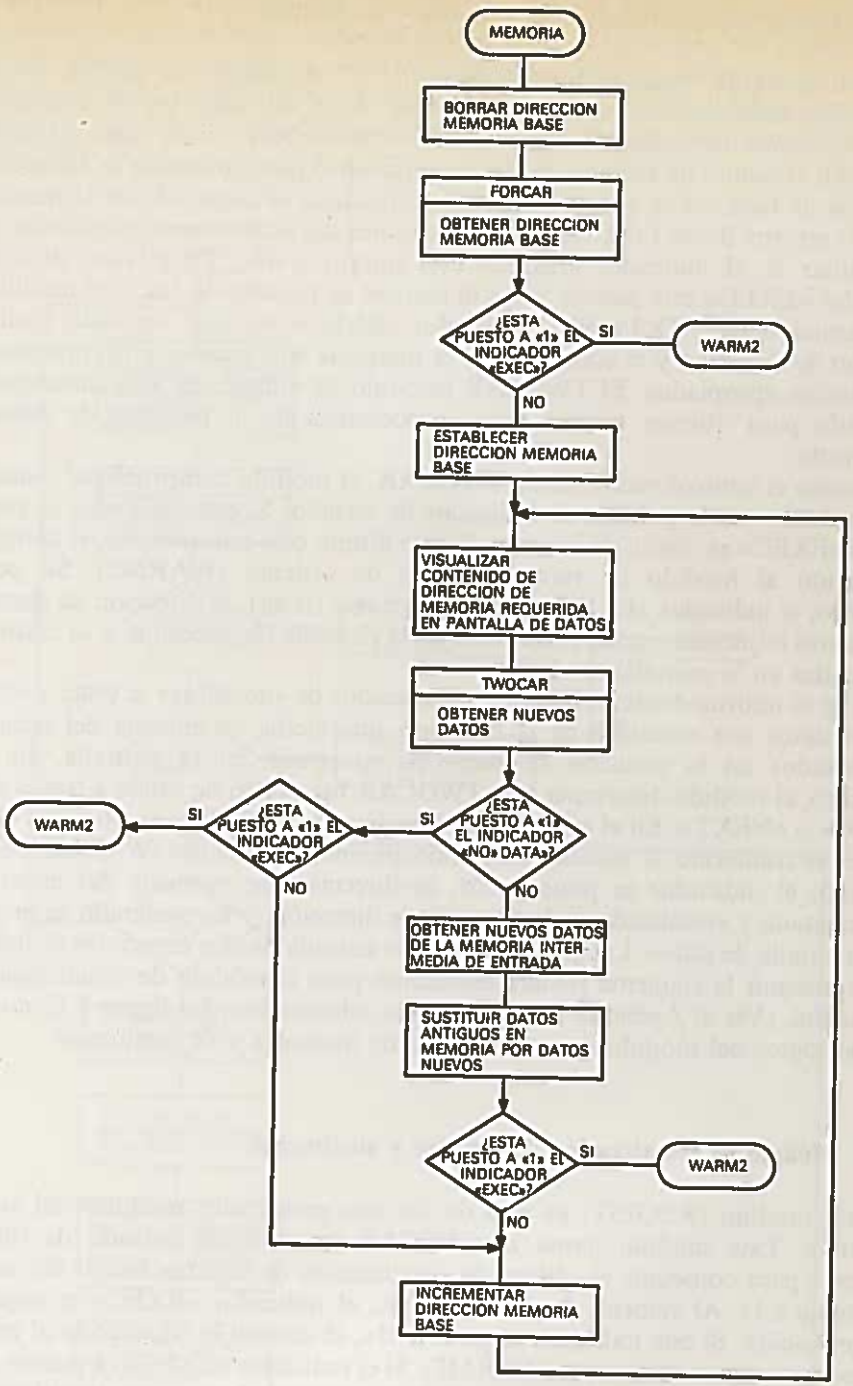


Figura 6.12 *Un diagrama de flujo del módulo de visualización de memoria y de sustitución (MEMORY).*

del área de conservación de registros y mostrados en el visualizador de dirección. En este momento, el módulo llama al TWOCAR para conseguir nuevos datos a partir del registro visualizado. Cuando el módulo de entrada de dos caracteres proporciona el control, el módulo determina el modo de ejecución examinando los indicadores del teclado. Si están puestos a «1» los indicadores de «no datos» y «EXEC», el control es transferido al módulo de reconocimiento de órdenes (WARM2). Si están puestos a «1» los indicadores de «no datos» y «NEXT», el índice de registro es incrementado y el contenido del registro se canaliza al visualizador adecuado.

Si el indicador «no datos» está puesto a «0», los nuevos datos del registro se obtienen a partir de la memoria intermedia de entrada del teclado y se almacenan

en la posición adecuada de conservación de registros. En este momento se comprueba el indicador «EXEC», y si está puesto a «1», el control se transfiere al módulo de reconocimiento de órdenes (WARM2). Si el indicador «EXEC» está puesto a «0», los datos del registro se visualizan y se procesa la pseudoinstrucción del usuario (ver Apéndice D para detalles adicionales). La figura 6.13 detalla el flujo lógico del módulo de visualización de registro y sustitución.

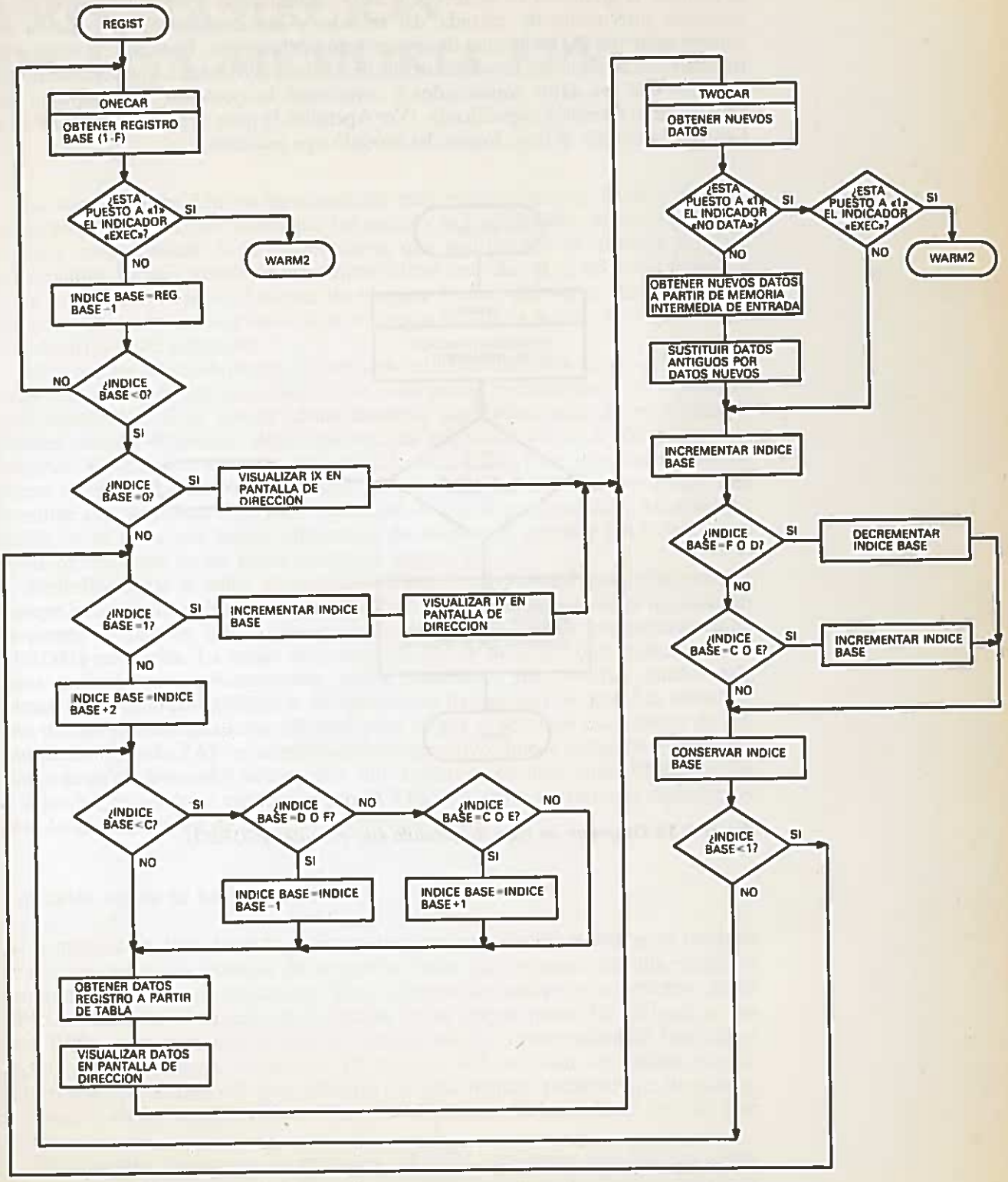


Figura 6.13 *Diagrama de flujo del módulo de visualización de registro y sustitución (REGIST).*

II.10. Módulo «go execute»

Este módulo (GOREQ) es el último de las tres principales funciones del sistema operativo. A la entrada (ver módulo de reconocimiento de órdenes), este módulo llama a FORCAR para conseguir la dirección en donde ha de comenzar la ejecución. Al retorno desde FORCAR, el indicador «no datos» se examina para determinar el modo de ejecución. Si este indicador está puesto a «1», el control se transfiere inmediatamente a RESTRT. Con ello se restauran los registros del Z80 y se reanuda la ejecución en la dirección de PC actualmente contenida a partir de la memoria intermedia de entrada del teclado y almacenada en la posición de conservación del PC en la zona de conservación del registro. Entonces el control se transfiere al módulo de reconocimiento de órdenes (WARM2), que restaurará los registros con los datos conservados y comenzará la ejecución del programa del usuario en la dirección especificada. (Ver Apéndice D para detalles adicionales.) La figura 6.14 detalla el flujo lógico del módulo «go execute».

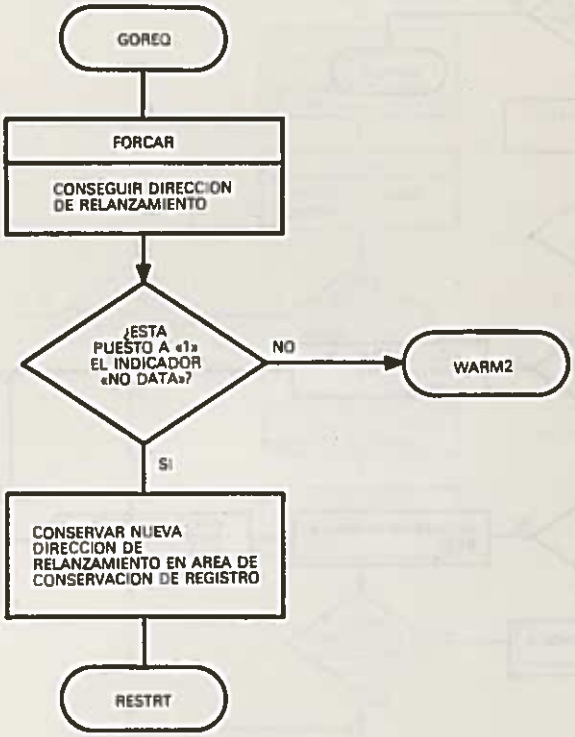


Figura 6.14 *Diagrama de flujo del módulo «go execute» (GOREQ).*

Capítulo 7

PROGRAMACION DE UNA MEMORIA EPROM

La computadora ZAP se ha concebido para resultar barata, fiable y fácil de construir. Para mantener al mínimo los costes y la complejidad, se han suprimido algunas características de la computadora que podrían ser de utilidad para un principiante. La más visible de las características omitidas es la del panel frontal y de la presentación visual. Aunque, de ninguna forma, ello va en detrimento del funcionamiento de la computadora, su inclusión facilitaría la comprobación inicial y el desarrollo del programa.

Para probar adecuadamente el ZAP, debe estar en memoria un programa. Este programa no ha de ser muy largo (sólo unas pocas instrucciones son necesarias para determinar si la computadora funciona adecuadamente). El problema se plantea cuando el usuario desea ejecutar un programa de 50 ó 100 octetos de longitud. Finalizamos con una situación de intríngulis. Para introducir efectivamente código-máquina en la memoria programable del ZAP, un programa que coordine esta actividad debe estar ejecutándose por el programador. Afortunadamente, si se tiene una forma alternativa de escribir el monitor ZAP de 1 K en EPROM, entonces ya no existe problema alguno.

Preferiblemente a dejar al experimentador a sus propias ideas, esta sección incluye información sobre la programación de EPROMs. Para resolver la situación de puesta en marcha, hemos ideado un diseño para un par de programadores de EPROMs manuales. La carga de programas en un programador manual es una tarea tediosa, pues, primeramente, están concebidos para rutinas mucho más cortas, tales como comprobación de operaciones de sistemas básicos. Sin embargo, una unidad manual puede modificarse para cargar el software de monitor de 1 K completo. Cuando ZAP es completamente operativo, puede utilizarle en conjunción con un programador automático. Ello ayudará a escribir varias EPROMs. En el caso de que no desee escribir su propia EPROM, debe consultar el Apéndice A para la disponibilidad de EPROMs preprogramadas.

Un rápido repaso de las EPROMs

A menudo se desea tener la característica «no volátil» una ROM, pero también se desearía tener las ventajas de lectura/escritura que proporciona una memoria programable de semiconductores. Una solución de compromiso efectiva es la EPROM. Esta es una memoria de lectura en su mayor parte. Es utilizada como una ROM para períodos largos de tiempo, siendo ocasionalmente borrada o reprogramada si fuera necesario. El proceso del borrado se realiza por la exposición del sustrato del chip, cubierto con una ventana transparente de cuarzo, a la acción de la luz ultravioleta. Veremos el proceso de borrado al final de este capítulo.

El elemento básico de la memoria EPROM, utilizado por Intel y otros fabricantes, es el tipo de carga almacenada llamado dispositivo de almacenaje de transistor FAMOS (semiconductor de óxido metálico de inyección de avalancha de puerta flotante). Aplicando selectivamente una carga de tensión de 25 V a

elementos direccionados, configuraciones de bits particulares que constituyen el programa pueden ser escritas en la EPROM. Esta carga, al estar rodeada por material aislante, puede durar años. Su exposición a una luz ultravioleta intensa puede drenar la carga y el resultado es el borrado de toda la información programada.

Hay muchas EPROMs en el mercado: las 2708, 2716 y 2732 son las más importantes. En su mayoría, los especialistas en computadoras han eliminado, por su dificultad de programación, a la 1702 y han optado por las más fáciles de programar: 2708 y 2716, con la ventaja adicional de su gran capacidad de almacenaje. Las EPROMs más recientes en el mercado son considerablemente más caras que las 2708. En vista de las consideraciones anteriores, la 2708 es la mejor compra. Siendo solamente un poco más cara, podría utilizar la 2758 para una alimentación única. Por estas razones, el programador de EPROM bosquejado en este capítulo es el 2708.

La figura 7.1 es un circuito del programador manual 2708. IC 5 y dos secciones de IC 3 proporcionan el impulso de programa de +25 V para las EPROM. El IC 5 está ajustado para una duración de 1 ms y es disparado por una transmisión de un nivel lógico «0» a un nivel lógico «1» en su entrada. La EPROM proporciona y absorbe corriente a través del terminal 18 de programación. Es necesaria una combinación de dispositivos mucho mejor que un simple controlador de colector abierto. En el modo de escritura, cuando el terminal 20 $\overline{CS}$/WE está a +12 V y entre impulsos de programación, el terminal 18 tiene que llevarse a nivel bajo por un dispositivo activo puesto que suministra una pequeña cantidad de corriente. El impulso de programación es, en sí mismo, de unos 30 mA y no puede ser fácilmente admitido sin una configuración en seguidor de emisor, Q1. Este impulso debe estar entre 25 y 27 V en el terminal 18. Bastará disponer de 3 pilas de 9 V. (Una alternativa es utilizar una fuente de alimentación comercial encapsulada de 24 V y 50 mA. Esta última alimentación puede estar provista de resistencias adecuadas para proporcionar la tensión deseada de 25 a 27 V.)

Para escribir un octeto en la EPROM, una dirección de 10-bit preestablecida en los interruptores (SW 1 hasta SW 10) designa cuál de los 1024 octetos recibirá los datos. Para comenzar en la posición 0, todos los interruptores estarán en la posición cerrada. Seguidamente, los 8 bits que han de almacenarse se establecen en los interruptores SW 12 hasta SW 19. Este octeto de datos debe aparecer en la presentación visual de salida LED 1 hasta LED 8. Finalmente, para poner el programador en el modo de escritura, el interruptor SW 11 se coloca en posición de abierto. La introducción real de datos se produce cuando se pulsa el botón pulsador de escritura PB 1. Este produce un impulso de 25 V con una duración de 1 ms en el terminal de programa del 2708. De acuerdo con las especificaciones del fabricante, ningún impulso de programación simple deberá durar más de 1 ms. Para la retención máxima de datos, se recomiendan 100 de estos impulsos de programación (totalizando 100 ms por octeto).

Lamentablemente, 100 ms no puede aplicarse a una dirección simple en su totalidad. Los fabricantes indican que debe realizarse secuencialmente y debe consistir en 100 aplicaciones de 1 ms. En síntesis, ello significa que para un programa de 25 octetos, cada dirección deberá ser escrita con un solo pulso y luego repetir el bucle 100 veces. Nunca he tratado de alargar el pulso y programar una 2708 más rápido que lo usual. La experiencia ha demostrado, sin embargo, que algunas EPROMs son escritas totalmente con apenas 2 ó 3 bucles. Obviamente, para una retención completa cada dirección deberá ser escrita de nuevo en un programador automático.

Leer de nuevo el contenido almacenado en una 2708 es fácil en el mismo programador manual. Primero, todos los conmutadores de entrada de datos, SW 12 hasta SW 19, están abiertos para el estado «1» y luego el conmutador de «lectura/escritura» SW 11 es colocado en la posición de cerrado o «de lectura». Ningún otro régimen pulsante o de reloj es necesario. La presentación visual de salida mostrará el contenido del octeto indicado por los conmutadores de entrada de dirección SW 1 hasta SW 10. Se mantendrá constante hasta establecer otra dirección. Leer los contenidos es simplemente una forma de incrementar esta dirección de 10 bits a través del margen de direcciones del programa.

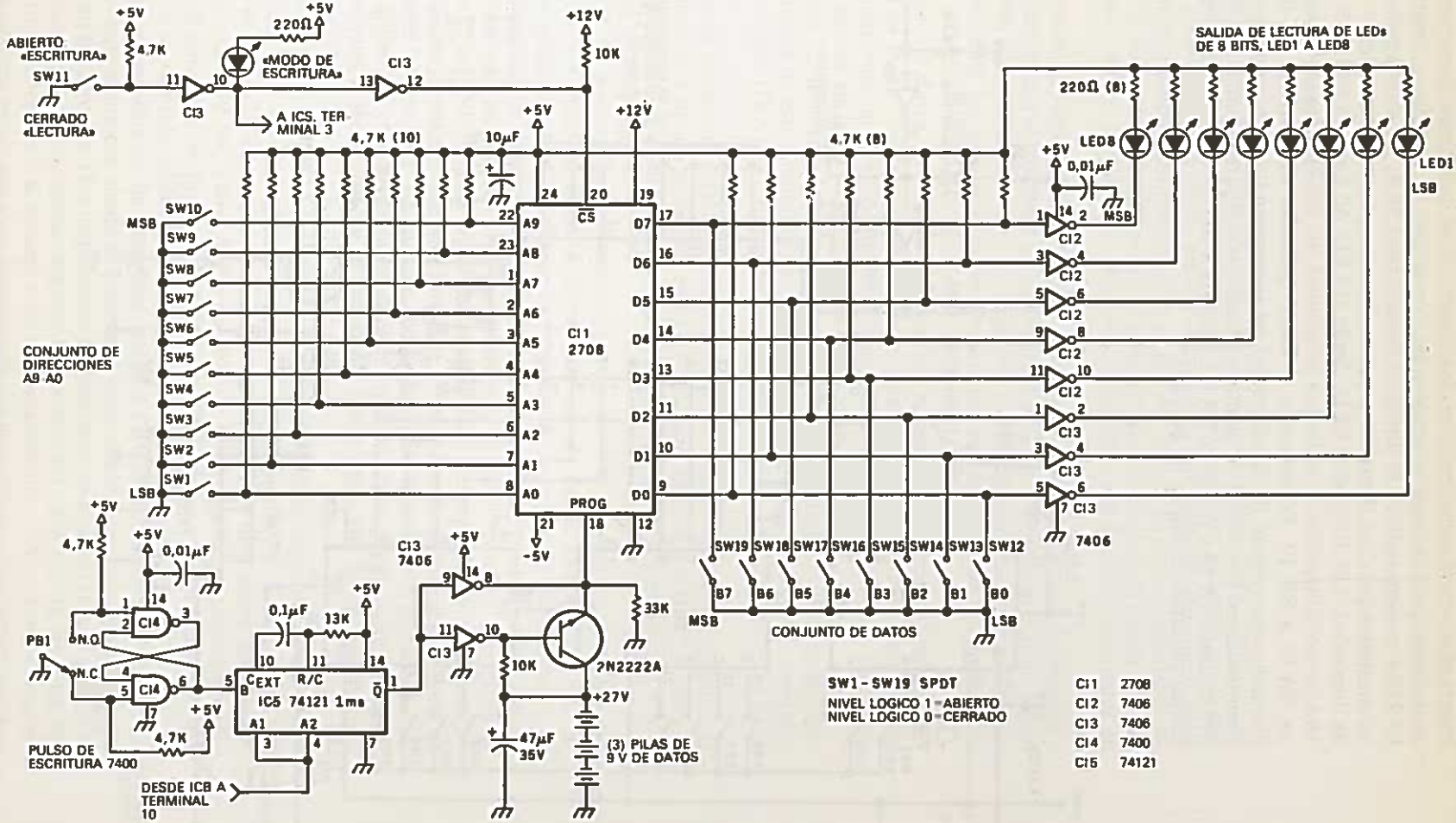


Figura 7.1 *Un diagrama esquemático de un programador manual 2708.*

Un programador manual un poco más complicado es mostrado en la figura 7.2. Tres contadores preajustables están insertados entre los conmutadores de entrada de dirección y la EPROM. En vez de cambiar las posiciones de los interruptores para cada dirección, se utilizan, ahora, solamente para preajustar los contadores para alguna dirección de comienzo. Si queremos programar un EPROM comenzando en la dirección hexadecimal 3AA, los interruptores deberán ser ajustados para esa dirección y pulsar el interruptor «address preset» (preajuste de dirección). Los 10 diodos LED, desde el LED A0 hasta el LED A9, leerían el 3AA como la dirección. Los datos a programar son establecidos en los conmutadores SW 12 a SW 19. Presionando el botón pulsador de escritura de datos PB 1 (llamado «incremento de direcciones»), se almacenan los datos procedentes de los conmutadores. Posiciones de memorias sucesivas son programadas con el ajuste de SW 12 hasta SW 19 y pulsando PB 1. La puesta a «0» del contador de direcciones se realiza pulsando el botón de borrado.

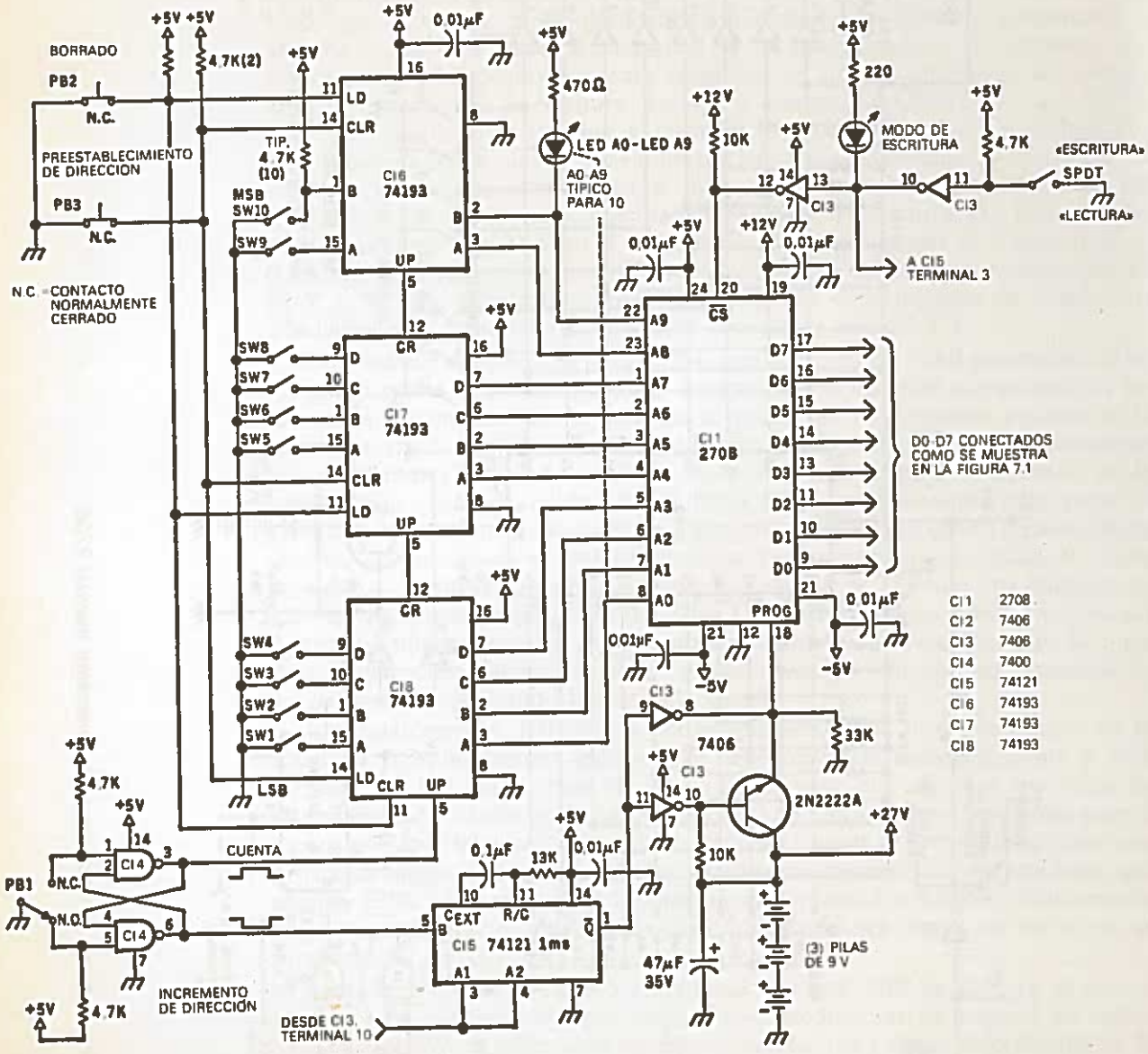


Figura 7.2 *Un diagrama esquemático de un programador manual 2708 autoincrementado. Diodos emisores de luz (LEDs) han de conectarse a las 10 líneas de entrada de dirección del 2708. Para mayor claridad, solamente un LED (conectado a la línea de dirección A9) es mostrado en el diagrama. Los otros diodos LED han de cablearse de la misma forma.*

Es fácil ver cómo este programador manual, aunque no mejore mucho el tiempo de programación, facilita la lectura de la memoria. Pone todos los interruptores de entrada de datos a un nivel lógico «1», pone el interfaz en el modo de lectura y preajustar y cargar una dirección de comienzo. La salida de lectura es realizada por la aplicación repetida del botón de incremento de la dirección.

Un programador automático

Necesitará una computadora ZAP en estado operativo para construir un programador automático. La complejidad del diseño puede ser reducida considerablemente aprovechándose de las señales «strobe» de E/S decodificadas, pero hasta el momento sin utilizar, proporcionadas en el ZAP básico. El circuito mostrado en la figura 7.3 utiliza 3 chips menos que el programador manual de la figura 7.2. Sus operaciones, aunque similares en el aspecto funcional, son bastante diferentes en detalle.

Cuatro señales «strobe» de E/S (port de entrada y salida 1 y port de entrada y salida 4) sincronizan el hardware y el software. La figura 7.4 muestra el flujo lógico para escribir una EPROM. Con la EPROM conectada directamente al bus de datos, solamente las señales «strobe» mejor que los registros de enclavamiento total son necesarias para este interfaz.

Para escribir datos, la secuencia deberá ser como sigue: Primero, una instrucción OUT 04 pulsa las líneas de borrado de contador de direcciones, poniéndolas a 0. Seguidamente, la EPROM se pone en el modo de programación, y el primer octeto se escribe en la EPROM con una instrucción OUT 01.

La figura 7.5 muestra cómo se selecciona el modo de programación de la 2708. La importancia de este circuito es que su salida está cableada como un convertidor digital-analógico (D/A) de 2 bits para controlar la línea de selección de chips de la 2708.

Cuando se ejecuta una OUT 04, el terminal $\overline{CS}$ verá 0 voltios que habilita el modo de lectura. Cuando se ejecuta una OUT 01, esta tensión será de 12 V para el modo de programa. Cuando no hay señal «strobe» presente, $\overline{CS}$ estará a +5 V y 2708 estará en triestado.

Un OUT 01 produce un pulso de programa de 25 V durante un milisegundo, mientras que los datos pertinentes estén en el bus de datos. Después de eso, un INP 01 es ejecutado, con lo que se incrementa el contador de direcciones a la siguiente posición de dirección. Realmente no estamos haciendo una función de entrada, sino que, en cambio, estamos utilizando la señal «strobe» decodificada de la instrucción INP 01 para significar «incremento del registro de dirección».

El hardware mantiene automáticamente el registro de la dirección, pero el software debe implementar sus propios contadores para mantener el registro de 0 a 1023 posiciones, así como el número de veces que se hayan programado los 1024 octetos completos. Recuerde que el fabricante recomienda 100 bucles de un milisegundo.

La lectura de la EPROM realizada automáticamente es algo muy simple. Un diagrama de flujo de la lógica es mostrado en la figura 7.6. El contador de dirección es borrado de nuevo haciendo una OUT 04. Los datos son leídos ejecutando una INP 04. Estos datos pueden ser almacenados y analizados. Finalmente, el contador de direcciones es incrementado de nuevo como una INP 01 y el proceso es repetido para leer el siguiente octeto.

Aunque el estudio se ha centrado en la EPROM 2708 de Intel como el dispositivo más ventajoso desde el punto de vista económico, hay muchas otras EPROMs en el mercado. Dos dispositivos de importancia especial (si sus precios y disponibilidad mejoraran cuando lea este libro) son las 2758 y 2716 de Intel. Se trata de EPROMs de 1 y 2 K de una sola alimentación de +5 V, respectivamente. La importancia de esto para el experimentador es que estos componentes pueden ser programados con un sencillo pulso de programa de 50 ms y de 25 V para cada dirección, que es mucho mejor que sucesivos bucles de 1 ms. Los tres circuitos de programador presentados han sido concebidos para 2708, pero pueden fácilmente

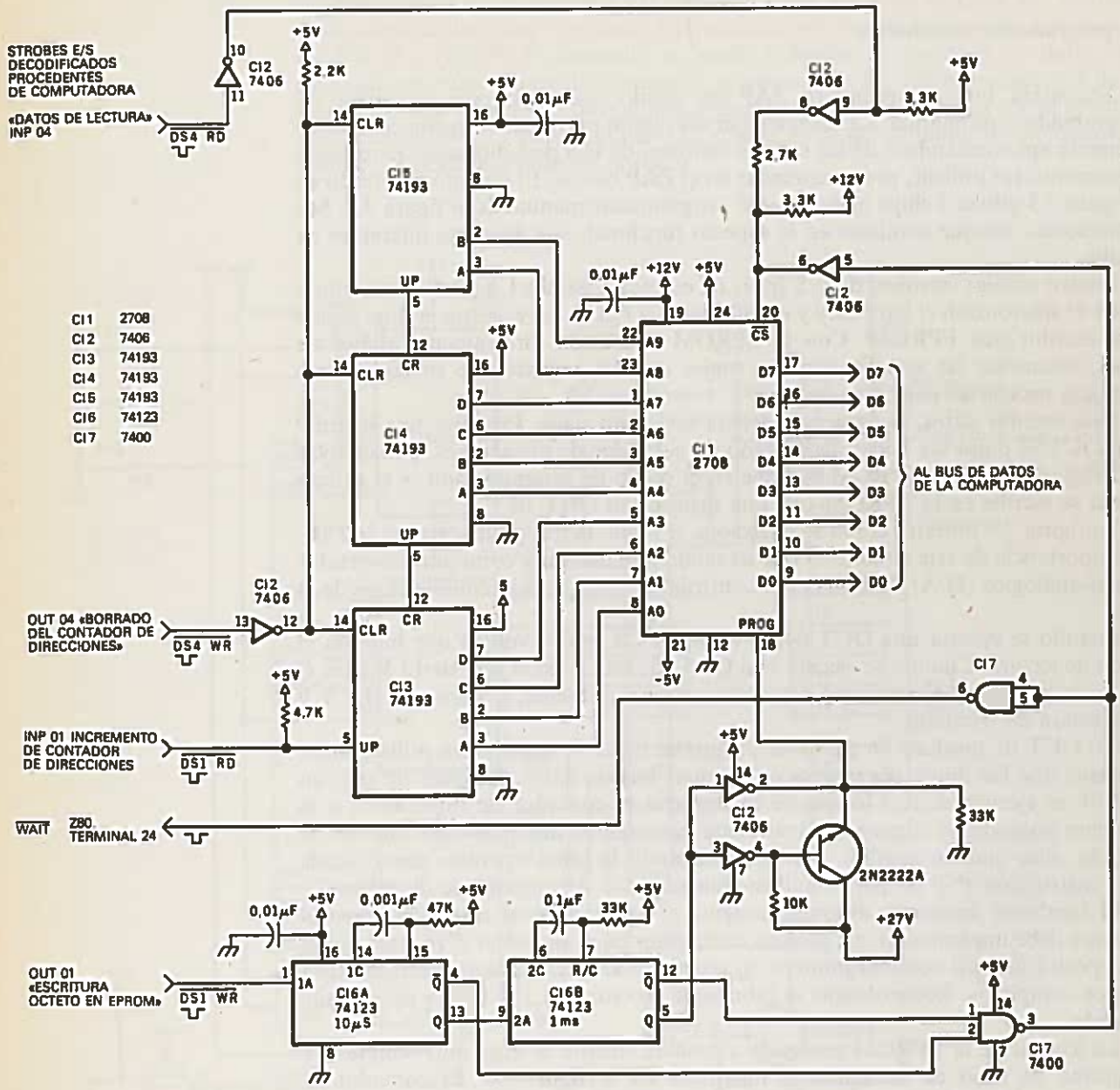


Figura 7.3 *Un diagrama esquemático de un programador automático 2708.*

ser adaptados para estos otros dispositivos. Cambiando el impulso temporizador de un solo disparo desde 1 a 50 ms, y cableando de nuevo unos cuantos terminales, se podrá completar la programación con una simple ejecución a través de las direcciones (no tiene que ser programada sucesivamente).

Borrado de una EPROM

Las memorias EPROM, compradas directamente de fábrica, se suministran totalmente borradas. Si planea escribir algún programa para la EPROM alguna vez, y no quiere modificarlo o no quiere cometer errores, olvídese del sistema de borrado. La mayoría de los que trabajan con computadoras querrán programar una EPROM. Y es, entonces, cuando se hace necesario conocer cómo borrarlos. Todos sabemos que la EPROM es borrada mediante radiaciones ultravioleta. Sin embargo, la duración, la distancia de la fuente de luz y la intensidad determinan la calidad del borrado.

Las personas encargadas de mantener las especificaciones del fabricante, durante la secuencia de programación, deberán también atenerse a los métodos precisos del borrado. A diferencia con el método de prueba del bucle de lectura después de escritura para programación, las EPROMs se suelen retirar del circuito durante el proceso de borrado. Por consiguiente, se aconseja realizar el proceso correctamente o tendrá que ser repetido.

La EPROM 2708 típica puede ser borrada por la exposición a luz ultravioleta de microondas de gran intensidad, con una longitud de onda de 2537 Å. La dosis

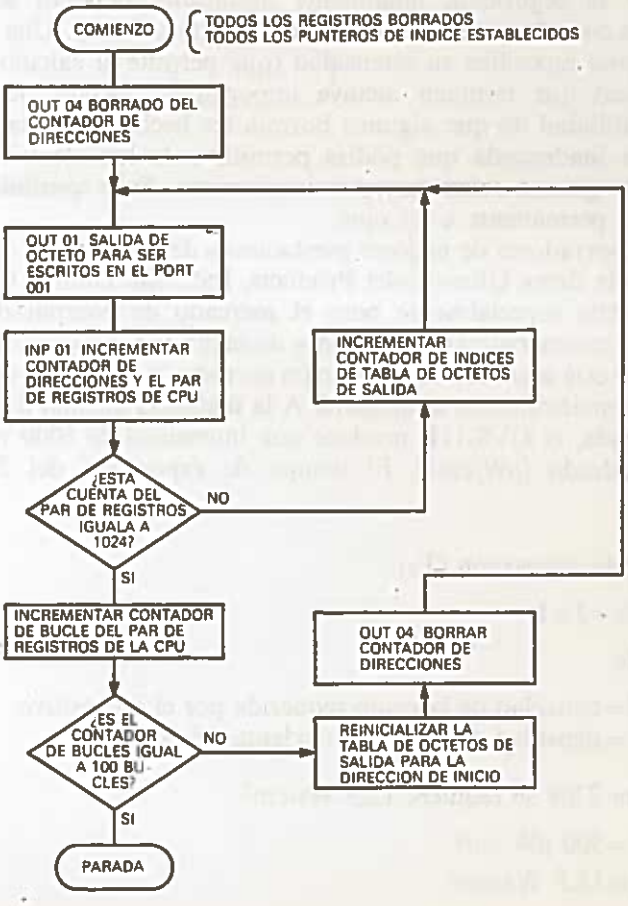


Figura 7.4 *Un diagrama de flujo de un ciclo de escritura de un programador automático de EPROM.*

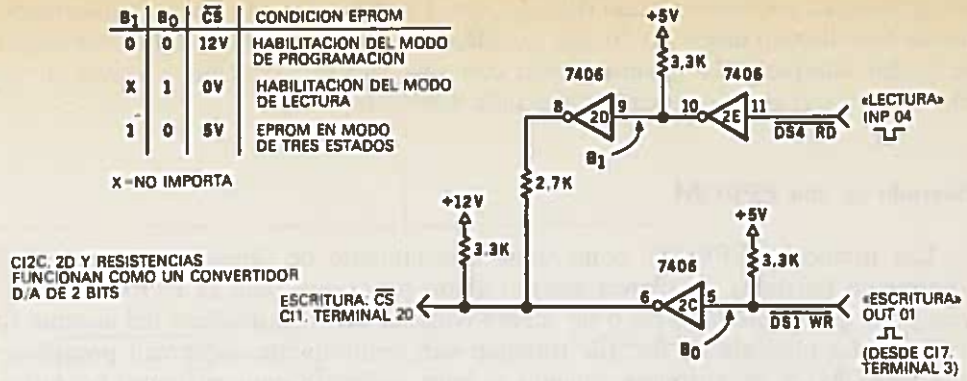


Figura 7.5 *Control programable de una línea $\overline{CS}$ de EPROM en un programador automático de EPROM.*

integrada recomendada (intensidad UV × tiempo de exposición) es 12,5 vatios por segundo y por cm² (vs/cm²). El tiempo requerido para producir esta exposición es función de la intensidad de la luz ultravioleta.

El coste y la seguridad, igualmente destacado, deberán ser los factores preponderantes cuando se selecciona un borrado ultravioleta. Una unidad comercial no solamente especifica su intensidad (que permite el cálculo del tiempo de exposición), sino que también incluye importantes enclavamientos. Hay que admitir la posibilidad de que algunos borradores hechos en casa pudieran tener una protección inadecuada que podría permitir a la luz ultravioleta escapar o accidentalmente girarse mientras está siendo vista. Tales posibilidades podrían causar un daño permanente a los ojos.

Uno de los borradores de mejores prestaciones de los que hay en el mercado es el UVS-11E de la firma Ultra-Violet Products, Inc., San Gabriel CA, 91776. Esta unidad está hecha especialmente para el mercado de computadoras caseras e incluye algunas características importantes de seguridad. La lámpara no funcionará al menos que esté asentada en la posición correcta, y si se eleva desde su bandeja de sujeción automáticamente se apagará. A la distancia normal de exposición que es de una pulgada, el UVS-11E produce una intensidad de 5000 microvatios por centímetro cuadrado (μW/cm²). El tiempo de exposición del 2708 se calcula fácilmente.

Tiempo de exposición (T_E)

$$T_E = J \div I$$

en donde

J = densidad de borrado requerida por el dispositivo
I = densidad de potencia incidente del borrado

Para una 2708 se requiere 12,5 Ws/cm²

$I = 500\ \mu W/cm^2$
$J = 12{,}5\ Ws/cm^2$

$$T_E = \frac{12{,}5}{5000 \times 10^{-6}} = 2500 \text{ segundos}$$

o T_E = 41,6 minutos para completar el borrado

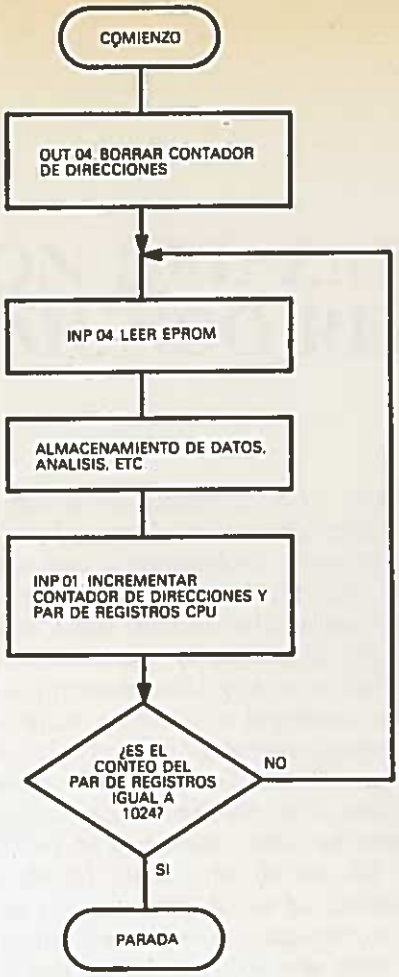


Figura 7.6 *Un diagrama de flujo de un ciclo de lectura de un programador automático de EPROM.*

Capítulo 8

CONEXION DEL ZAP CON EL MUNDO REAL

Ahora es evidente que la computadora ZAP puede configurarse de varias formas. Según sus necesidades, puede llegar tanto más allá del sistema básico que hemos descrito. Si desea una computadora personal que sea el equivalente a grandes sistemas de microcomputadoras comerciales, tiene que añadir bastante más memoria y periféricos. Debe permitirse la utilización de un sistema operativo más potente y, muy probablemente, el empleo de un lenguaje de alto nivel, tal como BASIC o Pascal. Si pretende utilizar la computadora ZAP como un sistema de procesamiento de palabras, entonces se requerirá una pantalla de vídeo y una impresora. A su vez, ello exige la adición de más «ports» (puertos) en paralelo y en serie. Cualquiera que sea la eventual configuración, no cambian las consideraciones de diseño que llevaron a la construcción de la computadora ZAP.

La computadora ZAP se ha concebido como un sistema experimentador. Este libro está estructurado de tal forma que deberá ser capaz de desarrollar una configuración del sistema y construirlo. No se ha considerado lo que lleva diseñar un sistema de procesamiento de palabras, o agregar un almacenamiento de disco flexible, porque queda fuera del objetivo de este texto introducirlo. El material necesario para cubrir adecuadamente esta materia sería suficiente para otro libro. Esto no significa, sin embargo, que todo esté terminado una vez que se haya construido la computadora ZAP y aprenda cómo escribir y ejecutar un programa corto. Por el contrario, una aplicación más significativa del ZAP es conectarlo a algo considerado como parte del «mundo real» y tenerlo funcionando en alguna tarea constructiva. La relación «potencia al peso» del ZAP le hace adecuado para aplicaciones de control inteligente. La verdadera clave de la utilización efectiva del ZAP es aprender cómo conectarlo con el mundo real.

Dentro del ámbito de los ejemplos directos que se han tratado, la computadora ZAP creada a partir de este libro debe ser una computadora de una sola tarjeta que se podrá utilizar con éxito en varias aplicaciones. Puesto que incluye un port en serie, dos ports en paralelo, un monitor de PROM y una memoria programable, el ZAP es, en muchos aspectos, equivalente a controladores digitales comerciales que cuestan cientos de dólares más.

Las computadoras pequeñas suelen utilizarse en adquisición de datos y en aplicaciones de control inteligente. Su función suele ser asimilar algunos parámetros de entrada y calcular un resultado. Por ejemplo, en un control de motor eléctrico de 100 HP, las entradas serían tensión corriente RPM, y la salida de control sería una tensión de corrección del factor de carga.

Con toda probabilidad, unos pocos de estos «controladores inteligentes» se utilizaron por la imprenta que imprimió este libro. Un lugar probable es la unidad de control electrónica que controla la densidad de la impresión y ajusta automáticamente el flujo de tinta. La computadora «lee» la impresión y decide si aumentar o disminuir el flujo de tinta para el papel. Esta decisión debe tener en cuenta varios parámetros de entrada, tales como la humedad, la temperatura, la velocidad del papel y el peso específico de la tinta. El algoritmo de control escrito en código máquina y almacenado en ROM se desplaza a través de todos los datos de entrada y genera su conclusión en la forma de una salida proporcional hacia una válvula de flujo de tinta.

En la mayoría de los casos las funciones computarizadas no se detienen con un simple error. En cualquier proceso en donde la capacidad repetitiva y el control de calidad sean importantes, parámetros de procesos significativos son constantemente controlados en su desviación con respecto a límites preestablecidos y si estos límites son excedidos suena una alarma. Para servir de ayuda en análisis a largo plazo, la función de adquisición de datos suele incluir la grabación de datos de procesos sin tratamiento a partir de los sensores de entrada de intervalos específicos y generar un registro permanente.

EL MUNDO REAL

No quiero confundirlo tratando demasiadas explicaciones comerciales de controladores de una sola tarjeta. Dudo que haya muchas prensas de papel continuo ocultas en lugares reservados a las que se desee añadir control por computadora. Hay, sin embargo, aplicaciones igualmente excitantes y menos esotéricas para controles de computadora de tipo casero. Algunas de las que podría mencionar en este momento serían la administración de la energía, la seguridad y la vigilancia ambiental: Denomino a estas aplicaciones sistemas del mundo real, en oposición al mundo de las computadoras TTL digitales.

Como el mundo real es todo lo que está fuera de la computadora, suele ser un ambiente analógico. La metamorfosis del ZAP en un controlador inteligente depende primariamente de su interconexión analógica efectiva. Por esta razón, el resto de este capítulo está dedicado al diseño y construcción de un interfaz de E/S analógico y económico.

Pero, primero, repasemos los fundamentos de la conversión D/A (digital/analógica) y luego tratemos de un método para utilizar un convertidor D/A para efectuar la conversión D/A (analógico/digital). En los sistemas de adquisición de datos, a menudo hay la necesidad de adquirir canales múltiples de alta resolución y entradas tanto de c.a como de c.c. Siendo éste el caso, examinaremos también un circuito que, en efecto, permite al ZAP funcionar como un voltímetro digital de 8 canales. Finalmente, como la relación temporal en muchos eventos es importante, el ZAP estará configurado como un reloj de tiempo real que indicara el tiempo en el cual se realicen las operaciones de control.

CONVERTIDORES DIGITAL/ANALOGICO

El convertidor D/A (digital/analógico) puede ser considerado como un potenciómetro programable controlado digitalmente que produzca una salida analógica. Este valor de salida (V_{OUT}) es el producto de una señal digital (D) y una referencia analógica (V_{REF}) y es expresada por la siguiente ecuación:

$$V_{OUT} = D\ V_{REF}$$

Para una gran extensión, ningún convertidor D/A o A/D es de gran utilidad sin la especificación del tipo de código usado para representar la magnitud digital. Los convertidores trabajan con códigos digitales monopolares o bipolares. El monopolar incluye el binario puro y el decimal codificado en binario (BCD). El binario desplazado, código Gray y complemento a uno o complemento a dos, se suele reservar para la operación bipolar. Sin embargo, nos limitaremos a las dos clases de binario: puro y desplazado.

Es importante recordar que la cantidad binaria presentada por la computadora es una representación de un valor fraccionario a multiplicar por una tensión de referencia. En fracciones binarias, el MSB (bit más significativo) tiene un valor de 1/2 ó 2^{-1}, el siguiente MSB es 1/4 ó 2^{-2} y LSB (bit menos significativo) es $1/2^n$ ó 2^{-n} (en donde n es el número de lugares binarios a la derecha del punto binario). La adición de todos los bits produce un valor que se aproxima a 1 (cuantos más bits, tanto más próximo estará dicho valor a 1). La diferencia algebraica entre el valor

binario que se aproxima a 1 y el valor 1 es el error de cuantización del sistema digital (a estudiar más adelante).

El binario desplazado es similar al puro con la salvedad de que el número binario 0 se establece para representar la cantidad analógica *negativa* máxima, el MSB es un 0 para valores analógicos negativos y un 1 para valores analógicos positivos.

La conversión de valores digitales en valores analógicos proporcionales es realizada por una u otra de dos técnicas básicas de conversión: el convertidor D/A de resistencias ponderadas y el convertidor D/A R-2R. El convertidor D/A de resistencias ponderadas es, hasta ahora, el circuito más simple y más directo. El decodificador en paralelo requiere solamente una resistencia por bit y trabaja como sigue: Los conmutadores son controlados directamente por las señales que representan el número digital D; corrientes con magnitudes de 1/2, 1/4, 1/8, ..., 1/(2^n) son generadas por resistencias con magnitudes de R, 2R, 4R, ..., 2^{n}R, que están conectadas por medio de interruptores entre una tensión de referencia, $-V_{REF}$, y el punto sumador de un amplificador operacional. Las diversas corrientes son sumadas y convertidas en tensión por un amplificador operacional (ver figura 8.1).

Aunque ésta pueda parecer una sencilla solución a un programa complejo, este método puede tener algunas ramificaciones potencialmente peligrosas. La exactitud de este convertidor es una función de las exactitudes combinadas de las resistencias, interruptores (todos los interruptores tienen alguna resistencia) y el amplificador de salida. En sistemas de conversión de resolución superior a 10 bits, las magnitudes de las resistencias resultan ser excepcionalmente grandes y el flujo de corriente resultante se reduce a un valor tan bajo que se pierde en ruido térmico del circuito.

Una alternativa razonable para el convertidor D/A de resistencias ponderadas es el convertidor R-2R. A menudo se le conoce como un convertidor D/A de resistencias en escalera y es el tipo más utilizado aun cuando utiliza más componentes. Este circuito (ver figura 8.2) también contiene una tensión de referencia, un conjunto de interruptores binarios y un amplificador de salida. La base de este convertidor es una red en escalera construida con dos valores de resistencia R y 2R.

Una resistencia (2R) está en serie con el interruptor de bits, mientras que la otra (R) está en la línea sumadora, de tal forma que la combinación forma una red en «pi» (π). Esto indica que las impedancias de las tres ramas de cualquier nudo son iguales y que la corriente I, que llega a un nudo a través de una rama, sale como I/2 a través de las otras dos ramas. Dicho de otro modo, una corriente producida cerrando un interruptor de bits es reducida a la mitad como si pasara a través de cada nudo en el camino hacia el final de la escalera. Simplemente expuesto, la posición de un interruptor, con respecto al punto en donde la corriente es medida, determina el significado binario del cierre del interruptor particular.

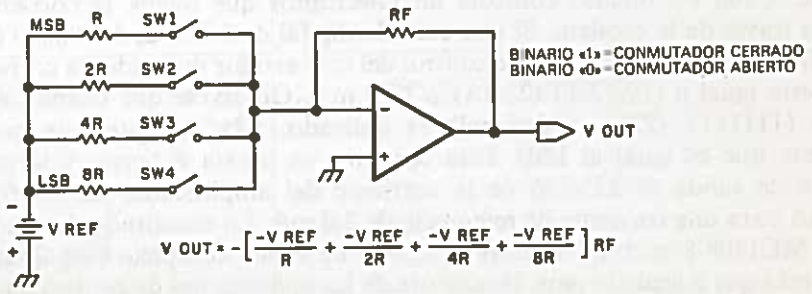


Figura 8.1 *Un convertidor D/A de resistencias ponderadas de 4 bits. Una palabra de 4 bits es utilizada para controlar cuatro interruptores monopolares y de una sola vía. Cada uno de estos interruptores está en serie con una resistencia. Los valores de la resistencia están relacionados como potencias de 2, como se muestra en la figura. Los otros lados de los interruptores se conectan juntos al punto sumador de un amplificador operacional. Corrientes con magnitudes inversamente proporcionales a las resistencias se generan cuando los interruptores están cerrados. Se suman por el amplificador operacional y se convierten en una tensión correspondiente.*

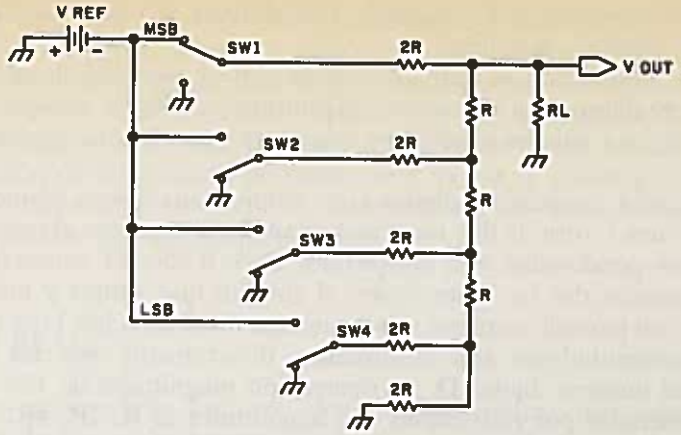


Figura 8.2 *Un convertidor D/A de resistencias en escalera R-2R de 4 bits. Este tipo de convertidor D/A hace uso de una red resistiva en escalera construida con resistencias de valores R y 2R. La topología de esta red es tal que la corriente que fluye en cualquier rama de un nudo de tres ramas se dividirá a sí misma igualmente a través de las dos ramas restantes. A causa de esto, la corriente se dividirá a sí misma por la mitad cuando pase a través de cada nudo, en su camino hacia el final de la escalera. Los cuatro interruptores están relacionados, de nuevo, por potencias de dos. La posición de cada interruptor con respecto a su distancia desde el final de la escalera determina su significación binaria.*

Este tipo de convertidor es fácil de construir puesto que sólo son necesarios dos valores de resistencia; de hecho, un solo valor R será suficiente si se utilizan tres componentes por cada bit. El mantener adaptados los valores de las resistencias con el mismo coeficiente de temperatura contribuye a un diseño muy estable. Ciertas soluciones de compromiso son necesarias entre los valores de las resistencias de la escalera y el flujo de corriente para equilibrar la exactitud y el ruido.

Una forma del circuito en escalera R-2R es el convertidor D/A multiplicador y está disponible con una referencia fija o con una referencia variable por medios externos. Los convertidores D/A multiplicadores que utilizan referencias analógicas variables externas producen salidas que son directamente proporcionales al producto de la entrada digital multiplicada por esta referencia variable. Estos dispositivos o tienen salida de corriente o de tensión. Los dispositivos de salida de corriente son mucho más rápidos porque no tienen amplificadores de salida que limiten el ancho de la banda; por tanto, tienden a costar menos que los del tipo de tensión.

Un convertidor D/A multiplicador de 8 bits económico es de Motorola MC1408-8 (ver figura 8.3). Como se mencionó anteriormente, este convertidor monolítico contiene una red en escalera R-2R y una lógica de conmutación de corriente. Cada bit binario controla un interruptor que regula la corriente que circula a través de la escalera. Si una entrada digital de 8 bits de 11000000 (192 en decimal) es aplicada a las líneas de control del convertidor ilustrado, la corriente de salida sería igual a (192/256) (2 mA) o 1,50 mA. Obsérvese que cuando el valor binario 11111111 (225 en decimal) es aplicado, hay siempre una corriente remanente que es igual al LSB. Esta corriente es puesta a tierra, y la máxima corriente de salida es 255/256 de la corriente del amplificador de referencia o 1,992 mA para una corriente de referencia de 2,0 mA. La exactitud relativa para la versión MC1408-8 es ± 1/2 el LSB o 0,19 % de escala completa (ver figura 8.4). Esto es más que adecuado para la mayoría de las aplicaciones de control analógico de computadoras a nivel casero.

El circuito final (figura 8.5) es un convertidor multiplicador D/A MC1408-8 de 8 bits. Como se ha bosquejado anteriormente, un «multiplicador» significa que utiliza una tensión de referencia variable externa. En este caso, una tensión regulada a través de un diodo zener de 6,8 V se aplica a través de una resistencia que establece la corriente que fluye al terminal 14 en una magnitud aproximada de 2 mA.

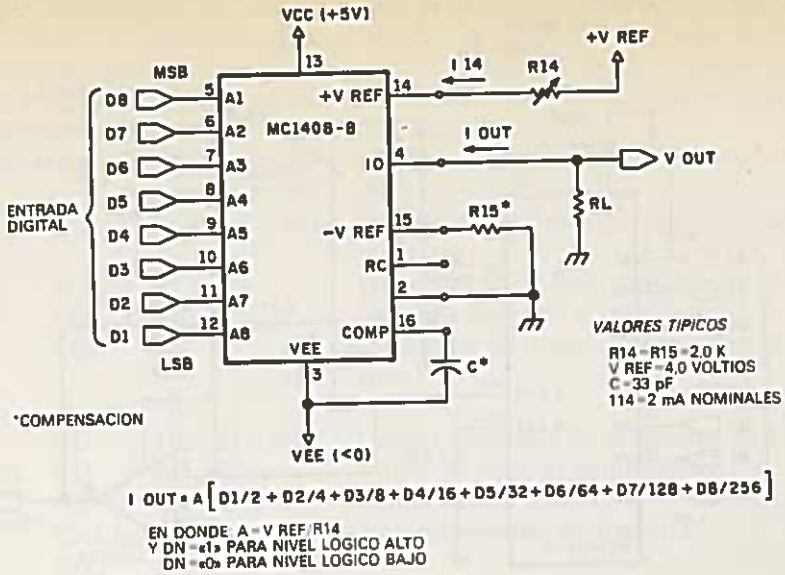


Figura 8.3 *Un típico convertidor D/A multiplicador monolítico de salida de corriente de 8 bits. Este circuito integrado de Motorola contiene una red R-2R como el de la figura 8.2, más lógica de conmutación de corriente adicional.*

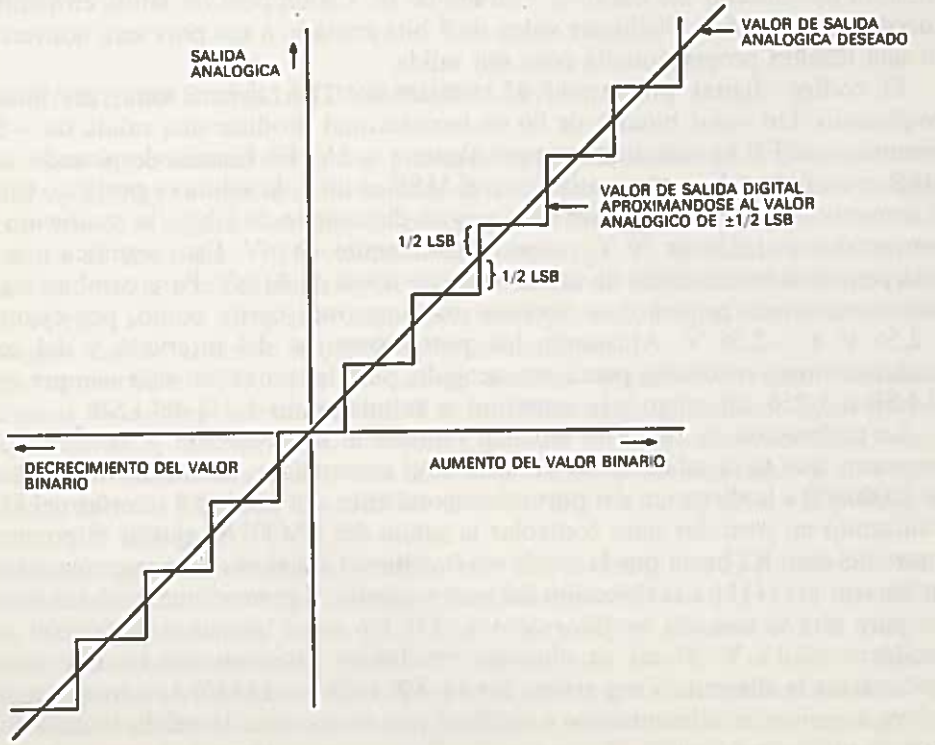


Figura 8.4 *Características de salida de un convertidor D/A mostrando la cuantificación menos significativa.*

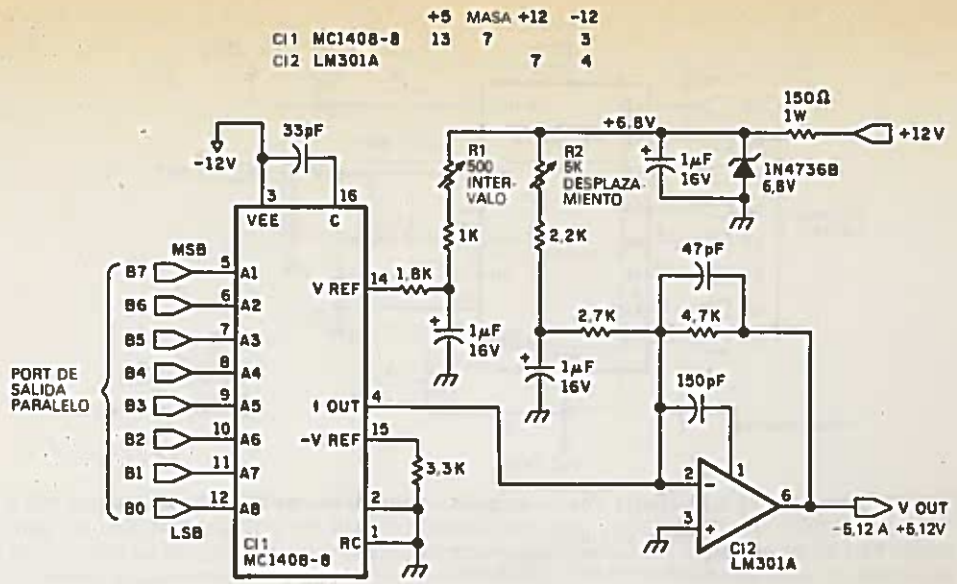


Figura 8.5 *Diagrama final de un convertidor multiplicador digital-analógico de 8 bits MC1408-8 con ajuste del intervalo y del desplazamiento.*

Una resistencia adicional R1 (también en esta rama de corriente) permite variar la corriente en un pequeño porcentaje y da oportunidad para ajustar el rango de la escala completa del convertidor D/A. La salida es una corriente que es equivalente al producto de esta corriente de referencia y los datos binarios en las líneas de control. La corriente es convertida a tensión a través del IC 9 y puede compensarse en cero mediante el empleo del potenciómetro de ajuste R2.

El empleo de este circuito con la computadora ZAP es simplemente una cuestión de conectar las líneas de entrada de IC 1 a un port de salida en paralelo adecuado en el ZAP. Cualquier valor de 8 bits enviado a ese port será convertido en una tensión proporcionada para esa salida.

El código digital presentado al convertidor D/A deberá estar en binario desplazado. Un valor binario de 00 en hexadecimal produce una salida de -5 V, mientras que FF hexadecimal es equivalente a $+5$ V. En binario desplazado, si el MSB es un 0, la salida es negativa, y si el MSB es un 1, la salida es positiva. Como el convertidor tiene un rango de 10 V y es un dispositivo de 8 bits, la resolución del convertidor es 1/256 de 10 V, o aproximadamente 40 mV. Esto significa que los más pequeños incrementos de salida serán de pasos de 40 mV. Para cambiar esto a incrementos más pequeños se necesita un rango más corto, como, por ejemplo, $+2{,}56$ V a $-2{,}56$ V. Ajustando los potenciómetros del intervalo y del cero, cualquier rango razonable puede ser escogido, pero la resolución será siempre igual al LSB o 1/256 del rango y la exactitud se estima como $\pm$ 1/2 del LSB.

La calibración es bastante sencilla. Aplicar la alimentación y, con un corto programa que da la salida de un valor desde el acumulador, enviar un valor binario de 10000000 a la dirección del port correspondiente a la placa del interfaz del D/A. Utilizando un medidor para controlar la salida del LM301A, ajustar el potenciómetro del cero R2 hasta que la salida sea 0 voltios. Con el mismo programa, cargar en binario 11111111 a la dirección del port y ajustar el potenciómetro del intervalo R1 para una lectura del medidor de $+5{,}12$ V. Un valor binario de 00000000 debe producir $-5{,}12$ V. Si no se obtienen resultados satisfactorios en este punto, desconectar la alimentación y retirar los c.i. MC1408-8 y LM301A; a continuación, volver a aplicar la alimentación y verificar que es correcta la salida binaria en el port de salida en paralelo. Nueve de cada diez veces, problemas como éste pueden atribuirse a la elección de un código de salida incorrecto.

Si la prueba es satisfactoria, estará en condiciones para generar salidas analógicas bajo el control del programa. Una prueba simple es diseñar una sección de memoria y secuencialmente extraer los valores para el D/A. Si la tabla tiene una

longitud de 250 octetos, con los valores fluctuando de 0 a FF hexadecimal a incrementos de 01, el resultado será una salida con una forma de onda de diente de sierra. Si las muestras se envían a la salida con suficiente rapidez y están conectadas a un altavoz, la forma de onda será audible. La frecuencia exacta será una función del bucle de temporización en actualización.

Lo siguiente es un programa corto que hace trabajar el D/A de esta forma:

```
START   EQU   0400        Dirección HL de comienzo de tabla de
                          memoria
END     EQU   05          Dirección H de final de tabla de memoria
OPORT   EQU   07          Número del port de salida de D/A
SAMP    EQU   A0          Constante de tiempo del régimen de mues-
                          treo

        LD    HL, START   Cargar dirección de comienzo de tabla
AGAIN   LD    A, (HL)     Valor de tabla al acumulador
        OURT  OPORT, A    Salida octeto a D/A
        CALL  DELY        Retardo tiempo de muestra
        INC   HL
        LD    A, H
        CP    END         Prueba para ver si se está al final de tabla
        JP    NZ, AGAIN   De no ser así, salida a la siguiente muestra
        HALT

DELY    LD    B, SAMP     Bucle de temporización del régimen de
                          muestreo
DCR     DEC   B
        JP    NZ, DCR
        RET
```

La tabla se puede poner a cualquier longitud. Los valores de la tabla se pueden calcular para producir cualquier configuración de formas de onda.

CONVERTIDORES ANALOGICO-DIGITALES

Siempre es conveniente estudiar primero los convertidores D/A. Son bastante sencillos y no hay un número abrumador de métodos de conversión. Introduciéndoles primero, se llegará a enterar del proceso de conversión binaria y apreciará los conceptos de resolución y de exactitud. Sin embargo, en términos prácticos, si fuera a utilizar la computadora ZAP para un modo de adquisición de datos (por ejemplo, para lectura y registro de temperaturas), necesitaría un convertidor A/D (analógico-digital) antes de uno D/A (digital-analógico).

Un convertidor A/D hace lo que su nombre indica. Convierte tensiones analógicas en una representación digital compatible con la entrada de la computadora. Como en el caso de un convertidor D/A de 8 bits, un convertidor A/D está sujeto a las mismas normas de conversión. Si se pretende leer una señal de 10 V con un convertidor de 8 bits, la resolución será 1/256 de 10 V (o 40 mV) y la precisión será de ± 1/2 LSB.

Cuanto mayor es la resolución tantos más bits se necesitan. El número de bits no establece el rango de un convertidor; sólo determina lo bien que se representa el valor. Un convertidor de 8 bits (A/D o D/A) puede prepararse con la misma facilidad para cubrir un rango de 0 a 1 V como para otro de 0 a 1000 V. Suele utilizarse la misma circuitería, pero se cambia una etapa de amplificación final o una red divisora de resistencias. Se entiende, por supuesto, que con un rango de 1000 V y un convertidor de 8 bits la resolución es de 4 V. Tal unidad sería inútil en señales de 0 a 10 V. El problema se puede reconciliar en varias formas, pero la solución más fácil es utilizar un convertidor con más bits. Un convertidor de 16 bits tiene 65 536 (2^{16}) pasos, en vez de 256 (2^8), cubriría el mismo rango de 1000 V en incrementos de 15 mV.

Para la computadora ZAP, la cuestión se hace más de funcionamiento a un precio razonable que un cierto grado de exactitud.

La conversión analógica-digital es considerablemente más cara que la D/A, pues el precio está directamente relacionado con la resolución y la exactitud. Hay muchas formas de realización de la conversión D/A. El rango oscila desde técnicas baratas muy lentas a otras caras ultra rápidas. Un convertidor A/D puede costar tan poco como 5 dólares o tanto como 10 000 dólares. Un convertidor A/D que explora sondas de termistor y proporciona datos para controlar la temperatura en un supermercado grande puede costar 4,75 dólares, pero no puede codificar información de vídeo procedente de un lector óptico.

El objetivo de este libro, por supuesto, es ayudar a construir su propia computadora; sirve de muy poco la presentación de diseños que están más allá del presupuesto razonable y de las capacidades medias de construcción. Por esas razones, he seleccionado entre una multitud de técnicas, cuatro diseños que pueden ser fácilmente construidos y conectados a través del interfaz en paralelo de la computadora ZAP. Uno de ellos deberá satisfacer sus necesidades básicas de adquisición de datos.

1. Convertidor básico analógico a ancho de pulso.
2. Convertidor contador de 8 bits de rampa binario de bajo coste y pequeña velocidad.
3. Convertidor de aproximaciones sucesivas de alta velocidad de 8 bits.
4. Interfaz de 8 canales, $3^1/_2$-dígitos 0—200 V c.a/c.c.

CONVERTIDORES CONTADORES BINARIOS Y ANCHO DE PULSO

Convertidor analógico a ancho de pulso

Este convertidor es uno de los codificadores más populares de bucle abierto por su simplicidad. Un diagrama de bloque básico se muestra en la figura 8.6. Este dispositivo utiliza un oscilador fijo en combinación con un circuito que genera un ancho de pulso que es una función lineal de la tensión de entrada analógica.

Para obtener esta duración de pulso lineal variable, los diseñadores frecuentemente utilizan un generador de rampa y un circuito disparador Schmitt. Se inicia un pulso de control de puerta en el comienzo de la rampa y un circuito de conteo comienza el incremento a la frecuencia determinada. Cuando la rampa lineal alcanza el mismo valor que la tensión de entrada, este conteo se termina. El valor que queda en el registro es representativo de la entrada analógica.

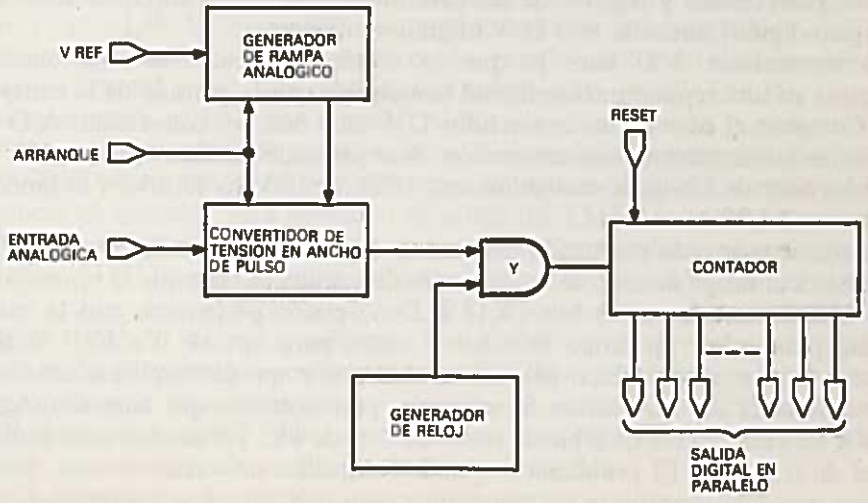
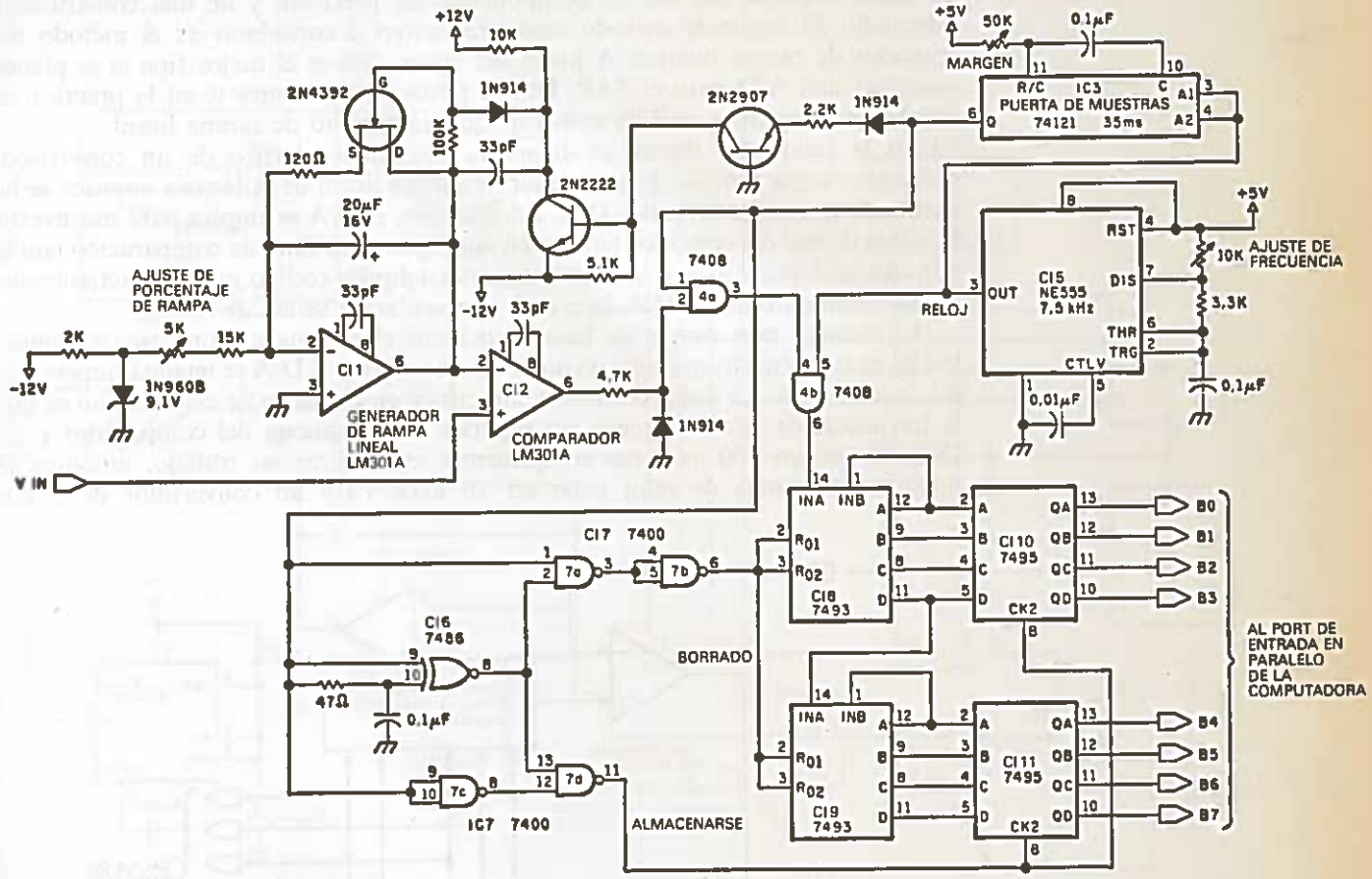


Figura 8.6 *Un diagrama de bloques de un codificador de analógico a ancho de pulso.*

La figura 8.7 es un esquema de un convertidor monopolar analógico a ancho de pulso que funciona con este principio. CI 1 está configurado como un generador de rampa lineal controlado por puerta e CI 2 es el comparador de entrada. El proceso comienza cuando la señal de reloj de 7,5 KHz activa a CI 3 (un 74121, monoestable) y comienza su período de 35 ms, que es el tiempo de la puerta. Al comienzo de este período de puerta, se genera un pulso que borra los dos 7493 y se activa el generador de rampa. Este, a su vez, habilita la señal de reloj al contador. La pendiente máxima de la respuesta del generador de rampa se establece a unos 10 V por 35 ms. El CI 2 compara, de forma continua, las tensiones de entrada y de rampa. Cuando son iguales, se interrumpe la señal de reloj hacia el contador y se reinicializa el generador de rampa. A la terminación del tiempo de puerta de 35 ms, sea cual sea el valor que esté en el contador se transfiere a un registro de almacenamiento de 8 bits. El valor almacenado en este registro es un número de 8 bits proporcional a la tensión de entrada. El proceso completo comienza de nuevo en el siguiente pulso de reloj.



CI#	TIPO	+5V	MASA	+12V	-12V
1	LM301A			7	4
2	LM301A	7			4
3	74121	14	7		
4	7408	14	7		
5	NE555	8	1		
6	7486	14	7		
7	7400	14	7		
8	7493	5	10		
9	7493	5	10		
10	7495	14	7		
11	7495	14	7		

NOTAS 1. AJUSTAR LA RAMPA PARA PASAR DE 0 V A LA ESCALA TOTAL DURANTE EL TIEMPO OPERATIVO DE LA PUERTA DE MUESTRAS

2. AJUSTAR LA FRECUENCIA PARA PRODUCIR EL CONTAJE DESEADO. SEA CUAL FUERE. PARA REPRESENTAR LA TENSION DE ENTRADA, ESTO ES. 256 CONTEOS DURANTE EL PERIODO DE MUESTRA PARA 2.56 VOLTIOS

Figura 8.7 *Diagrama esquemático de un convertidor monopolar analógico a ancho de pulso.*

Seleccionando adecuadamente los tiempos de puerta y la frecuencia de reloj, se pueden cambiar el intervalo funcional y la resolución del circuito. Con un tiempo de puerta de 35 ms y una frecuencia de reloj de unos 7500 Hz se deben contar 256 pulsos de reloj durante el tiempo de puerta. El potenciómetro de ajuste de la temporización de la rampa debe regularse de modo que el contador alcance un conteo máximo cuando se aplique 2,56 V a la entrada de CI 2. Un divisor 10:1 unido a esta entrada permitirá que el mismo conteo de 8 bits represente a 25,6 V.

Este circuito es simple, pero su exactitud depende de la estabilidad de las secciones individuales del mismo. Para usarlo, se conectan las salidas del registro a un port de entrada en paralelo. Basta con la lectura del port cuando se quiera el valor más reciente. El circuito se actualiza automáticamente 28 veces en un segundo, por lo que ninguna lectura será superior a 35 ms.

Convertidor contador de rampa-binario

La anterior técnica de A/D se utiliza con mucha frecuencia en medidas de alta precisión con pequeñas velocidades de muestreo. Lograr estos resultados, no obstante, depende del uso de componentes de precisión y de una construcción adecuada. El siguiente método más productivo a considerar es el método del contador de rampa binario. A juicio del autor, este es el mejor tipo si se planea construir una A/D para el ZAP. Utiliza pocos componentes y, en la práctica, es mucho más rápido y fácil de construir que un circuito de rampa lineal.

En la figura 8.8 ilustra el diagrama de bloques básico de un convertidor contador rampa binario. El generador de rampa lineal de la técnica anterior se ha sustituido por un convertidor D/A. En este caso, el D/A se emplea para reconvertir la salida digital del contador binario en analógica para fines de comparación con la entrada analógica. Si son iguales entonces cualquier código que esté actualmente activo en la entrada de D/A lo estará también en la salida de A/D.

La manera más simple de hacer funcionar el sistema es comenzar el conteo inicial en 0 y permitir que siga dicho conteo hasta que el D/A se iguale o supere a la salida analógica. La única consideración crítica en el diseño de este circuito es que la frecuencia de reloj no pueda ser superior a la respuesta del comparador y del D/A. Si tardan 100 μs estos componentes en realizar su trabajo, entonces la máxima frecuencia de reloj debe ser 10 kHz. Para un convertidor de 8 bits

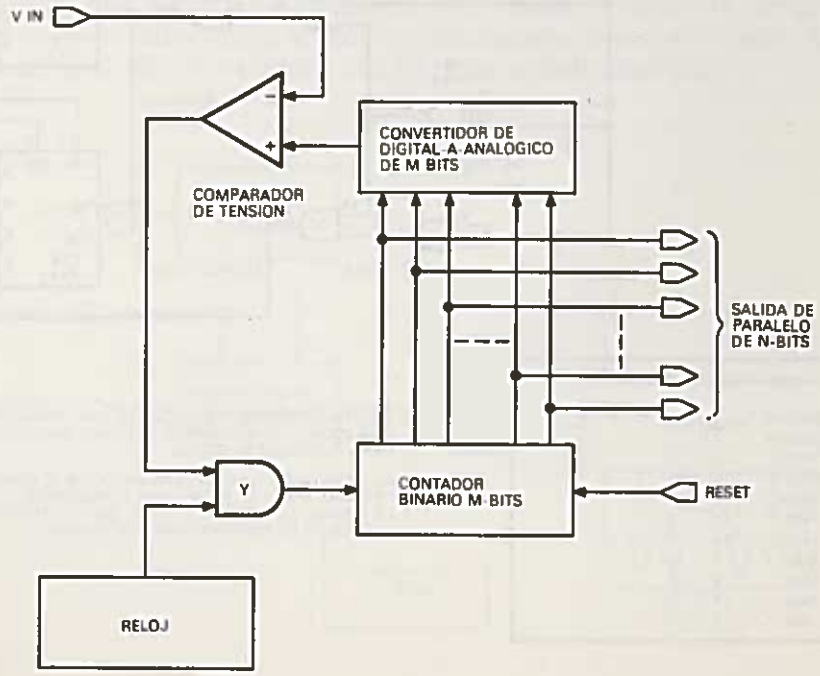
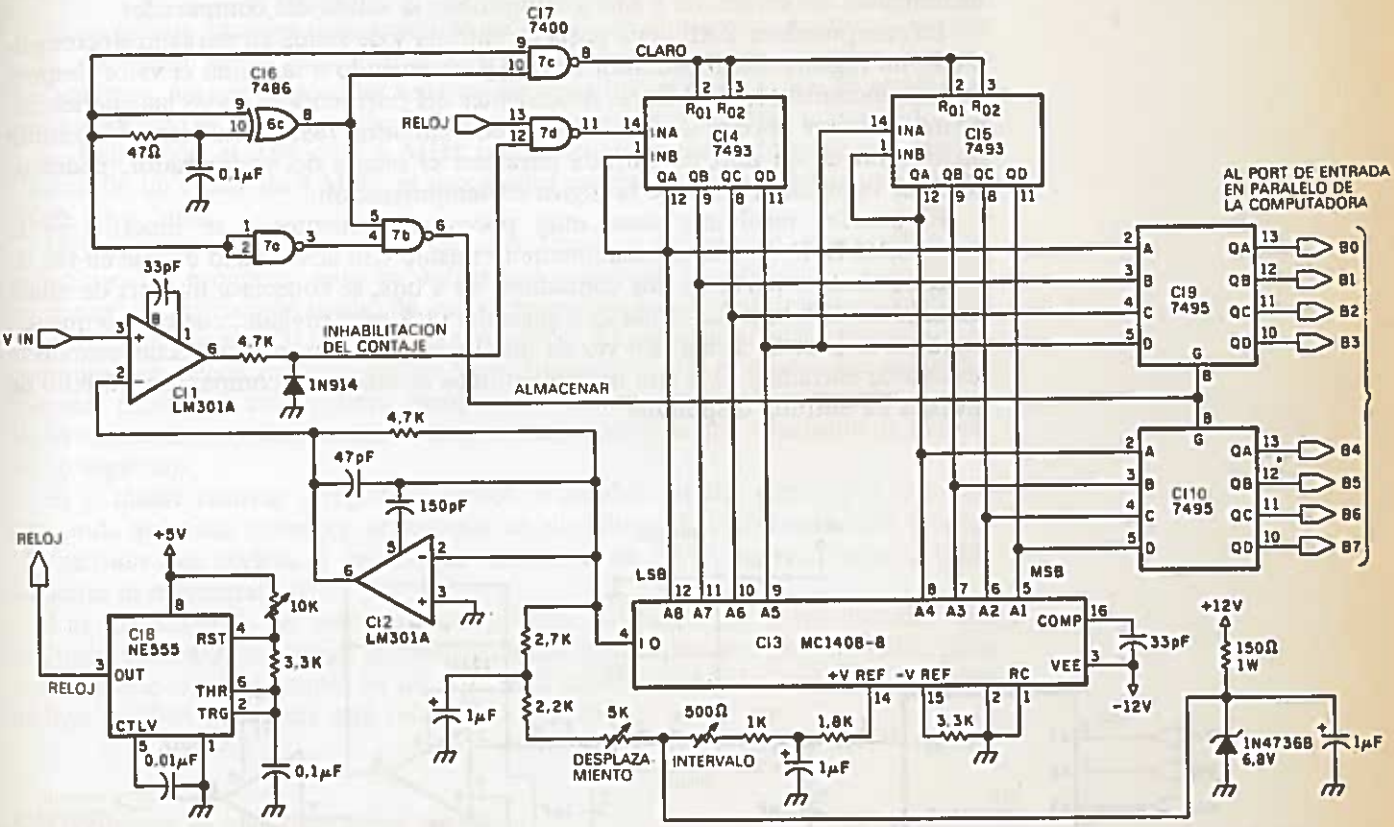


Figura 8.8 *Un diagrama de bloques del convertidor contador A/D de rampa binario.*

(contando desde 0 hasta 256 en cada periodo de muestra), la velocidad máxima de las muestras es 10 000/56 ó 39 muestras por segundo. En la práctica, sin embargo, 5 μs es el tiempo de ajuste más razonable, que corresponde a unas 750 muestras por segundo. Para velocidades todavía más altas, utilizamos una clase diferente de A/D (que estudiaremos más adelante).

La figura 8.9 muestra un esquema de un convertidor contador de rampa binario que utiliza un chip de convertidor D/A multiplicador MC1408-8. La salida del contador se conecta al MC1408-8 para proporcionar una comparación directa de realimentación analógica del valor establecido en el contador. Inicialmente, los circuitos CI 4 e CI 5 están borrados y la salida de D/A deberá ser igual a la tensión de entrada mínima. Para un convertidor de 0 a 5,12 V ésta sería de 0 V. Para una unidad de $-2{,}56$ a $+2{,}56$ V sería $-2{,}56$. Si la salida de IC 1 es menor que V_{IN}, se permite que los pulsos de reloj alcancen el contador. Como cada pulso incrementa el contador, la salida del D/A mantiene la elevación hasta que, a la larga, sea igual o apenas supere el valor de V_{IN} en el comparador. Cuando ello sucede, se inhiben los pulsos adicionales de reloj. Al final del periodo de muestra, el valor del conteo de CI 4 e CI 5 se almacena en un registro separado. Para que el ZAP lea estos datos, se requiere conectar este registro a un port de entrada y leerlos directamente.



CI#	TIPO	+5V	MASA	+12V	-12V
1	LM301A	7			4
2	LM301A			7	4
3	MC1408-8	13	7		3
4	7493	5	10		
5	7493	5	10		
6	7486	14	7		
7	7400	14	7		
8	NE555	8	1		
9	7495	14	7		
10	7495	14	7		

Figura 8.9 *Un diagrama esquemático de un convertidor contador A/D de rampa binario de 8 bits.*

Empleo de la computadora para sustituir al contador

La figura 8.9 es un circuito autónomo. No precisa de la computadora para funcionar. El A/D se actualiza por sí mismo a la velocidad de muestreo preseleccionado y carga este valor en un enclavador («latch») de 8 bits. En lo que respecta a la computadora, hay una lectura de régimen permanente a partir del convertidor. Cada función requerida para realizar la conversión A/D es construida a partir de componentes de hardware.

Hay ciertas ventajas en este método. El A/D puede ser montado y probado independientemente sin una computadora. Por ejemplo, se puede aplicar una tensión a la entrada y el valor de 8 bits se puede visualizar con 8 LEDs. Se ha presentado en este libro la posibilidad de probar independientemente cada subsistema. Si se cree tener cierta maestría en el arte de programar y que es mejor no construir interconexiones complicadas, se puede sustituir gran parte del hardware de la figura 8.9 por subrutinas de software.

Considere, por un momento, los principales elementos de diseño. Este A/D de 8 bits tiene cuatro secciones: D/A, comparador analógico, contador de 8 bits y lógica de temporización. El comparador analógico y la escalera de resistencias son componentes necesarios, pero las dos últimas secciones son primordiales para la síntesis a través de la computadora. La función combinada de estos dispositivos es incrementar un conteo de 8 bits y comprobar la salida del comparador.

La computadora ZAP tiene ports de entrada y de salida en paralelo. Incrementando un registro del procesador central y obteniendo a la salida el valor después de cada incremento, las 8 líneas procedentes del port tendrán todas las apariencias de un contador normal de 8 bits hecho con circuitos 7493 y similares. Utilizando un solo bit de un port de entrada para leer el estado del comparador, podemos sustituir también el resto de la lógica de temporización.

El interfaz resultante tiene muy pocos componentes y se muestra en la figura 8.10. El D/A tiene esencialmente lo mismo con la salvedad de que en vez de ser controlado a partir de dos contadores de 4 bits, se conecta a un port de salida en paralelo de 8 bits. La salida analógica del D/A será enviada, cualquiera que sea su valor, al port de salida. En vez de una lógica cableada para detectar cuándo la tensión de entrada y D/A son iguales, unimos la salida del comparador al bit 0 de un port de entrada disponible.

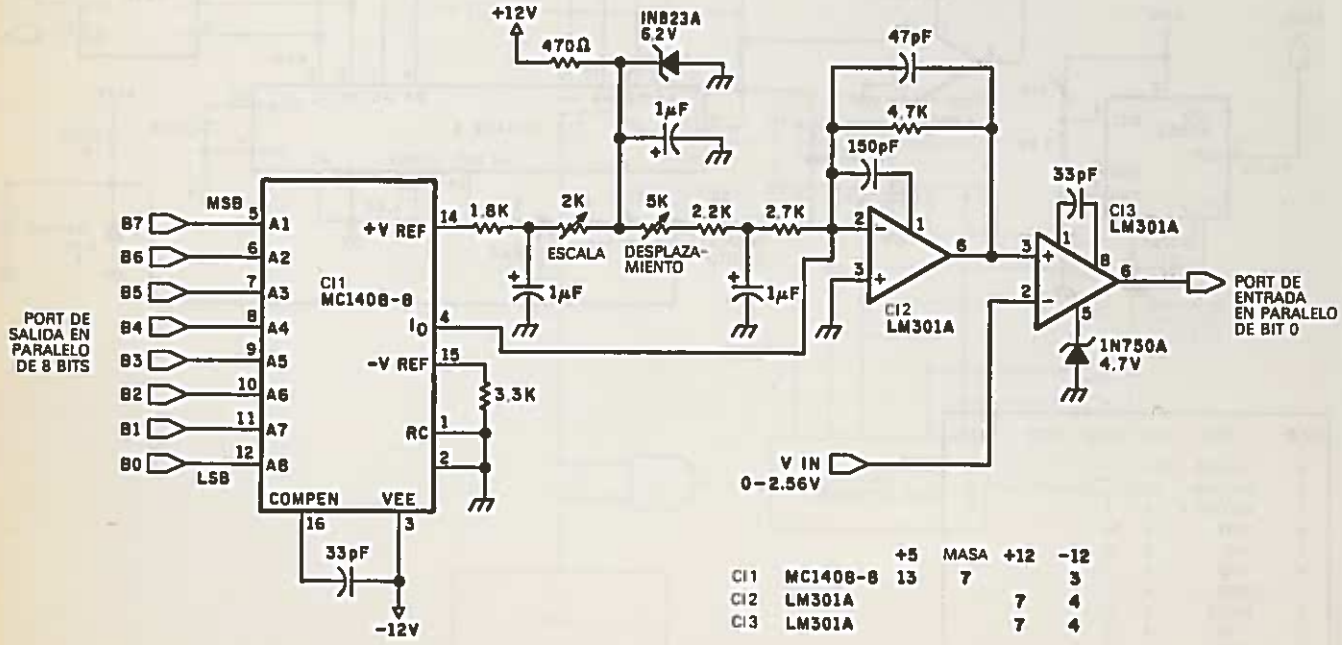


Figura 8.10 *Un convertidor A/D de 8 bits controlado por software.*

El proceso de conversión no es diferente a la versión de hardware. Primero borramos un registro (B, por ejemplo) y luego se saca el valor del registro hacia el port unido al D/A. Este pondrá el D/A a su salida mínima. Seguidamente, se lee el port de entrada que tiene el comparador unido y se revisa el bit 0 (un nivel lógico «1» indica que las tensiones de entrada y de D/A son iguales). Si el comparador está a nivel bajo (las tensiones no son iguales), el registro será incrementado y el proceso se repetirá. El registro será incrementado al punto en que la salida D/A y la tensión de entrada desconocida sean iguales. El comparador será entonces conmutado. En este momento, el programa se detiene y el valor del registro B es el equivalente digital a la tensión de entrada. El programa para realizar esto es el siguiente:

```
        MVI   B       Borrar registro B
        OUT   0, B    Salida del registro B
AGAIN   INC   B       Incrementar registro B
        OUT   0, B    Salida registro B
        IN    04      Leer port del comparador
        ANA   01      Aislar bit 0
        JNZ   AGAIN   Continuar si las tensiones no son iguales
        HLT           Valor de A/D está en registro B
```

El programa anterior debe repetirse cada vez que se necesite una nueva lectura y la velocidad de la muestra se podrá ajustar dentro de amplios límites. Recordemos, sin embargo, que todavía tenemos que esperar a que se establezca adecuadamente la circuitería de D/A y no se deberá incrementar más rápido que 5 µs. Utilizando el Z80 de 2,5 MHz no se deberá presentar problema alguno. Utilizando un cristal de 4 MHz, el procesador central podría necesitar unos pocos NOPs en el bucle.

Hay muchas variaciones en este circuito. Como se ha descrito, se necesitan 255 iteraciones del programa para encontrar una respuesta. En una computadora con un tiempo de instrucción medio de 2µs, el programa podría tardar 3µs en finalizar, limitándonos a unas 300 muestras por segundo. Al agregar las otras tareas que la computadora puede realizar, podría verse limitado a 100 muestras por segundo. Ejecutar rutinas de conteo lleva tiempo; sin embargo, no será un problema si se vigila correctamente una sonda de temperatura que tiene una constante de tiempo de 30 segundos.

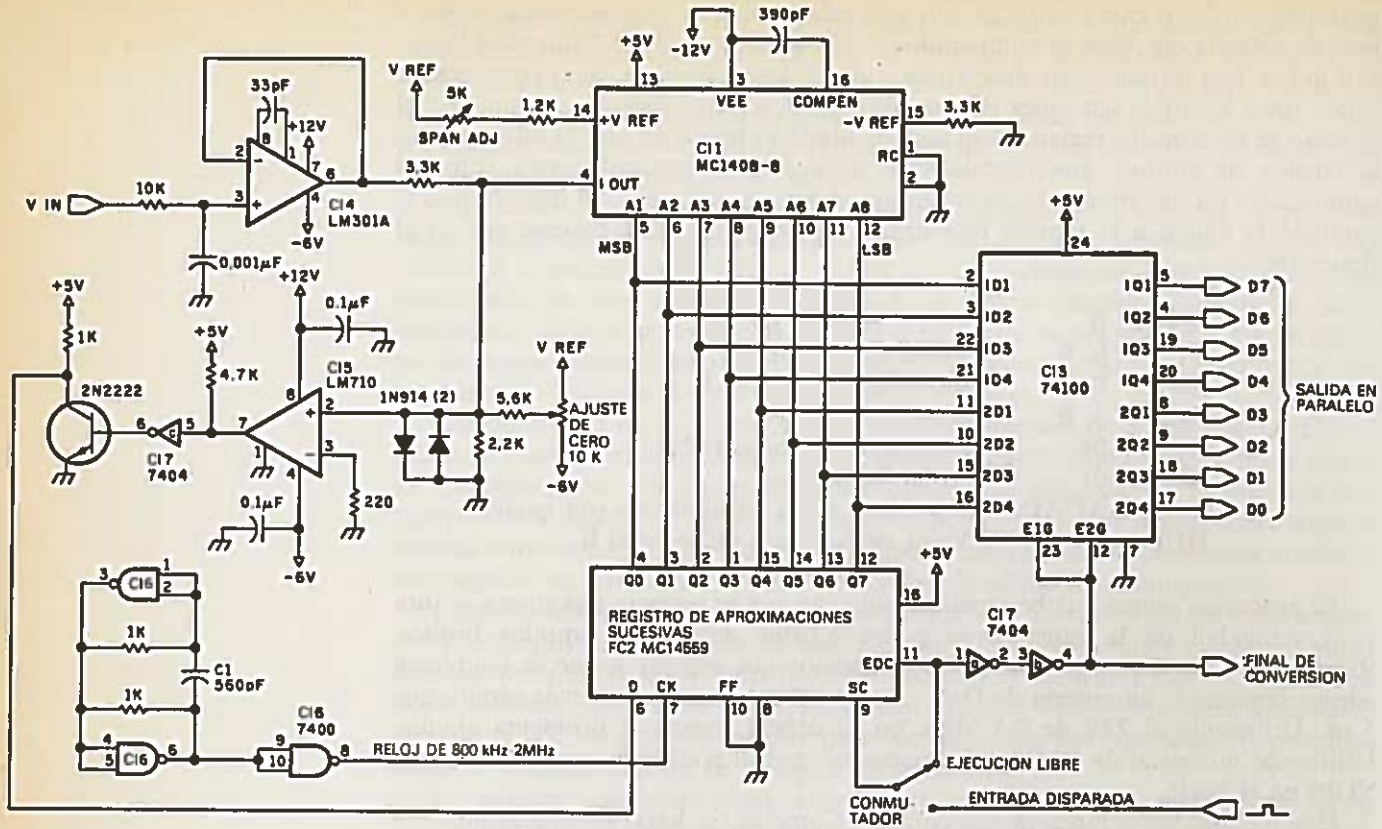
Si se quiere rastrear y registrar señales de cambio rápido, como, por ejemplo, una onda acústica, entonces se requiere un algoritmo de conversión más rápido. Un método que acelera el proceso se denomina de aproximación sucesiva (más adelante lo estudiaremos).

Las capacidades de este circuito pueden ampliarse de otras formas. Un multiplexor CMOS adicional se puede conectar a 3 bits de otro port de salida para cambiar este circuito simple en un A/D de 8 canales. Además, como ese circuito incluye un D/A, su salida está también disponible.

Convertidores de aproximaciones sucesivas

Es muy probable que uno de los tres convertidores presentados hasta ahora será suficiente para la adquisición de datos no crítica. Señales de cambio lento pueden tratarse de forma exacta y eficaz. Sin embargo, hay ocasiones en que la señal en cuestión no es lenta, ni lleva un transitorio particular que deba ser capturada. Por ejemplo, para detectar un evento de 100µs se requiere un convertidor con una capacidad de 20 000 muestras por segundo. En tales casos, necesitamos un método de conversión mucho más rápido.

La figura 8.11 es un esquema de un convertidor de 8 bits de alta velocidad para propósitos generales. Es capaz de trabajar con velocidades de muestreo de más de 200 000 muestras por segundo. Para alcanzar estas velocidades se utiliza la técnica llamada de aproximaciones sucesivas. Al igual que el convertidor contador de rampa binario, este A/D también incorpora un D/A en un bucle de realimentación,



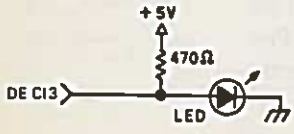
CI#	TIPO	+5V	MASA	+12V	-12V	-6V
1	MC1408-8	13	1		3	
2	MC14559	16	8			
3	74100	24	7			
4	LM301A			7		4
5	LM710			8		4
6	7400	14	7			
7	7404	14	7			
8	MC1408-8	13	1		3	
9	LM301A			7	4	

NOTAS 1. TODAS LAS RESISTENCIAS SON DE 1/4W, 5 %, A NO SER QUE SE INDIQUE ALGO EN SENTIDO CONTRARIO.

2. TODOS LOS CONDESADORES SON DE CERAMICA DE 100 V. A NO SER QUE SE INDIQUE ALGO EN SENTIDO CONTRARIO.

3. CON LOS COMPONENTES MOSTRADOS, LA FRECUENCIA DEL RELOJ ES 100 kHz. ESTO ES, 100 000 CONVERSIONES POR SEGUNDO EN MODO DE EJECUCION LIBRE.

4. EL SIGUIENTE CIRCUITO PUEDE SER AGREGADO A CADA TERMINAL DE SALIDA DE CI 3, SI SE DESEA UN INDICADOR VISUAL.



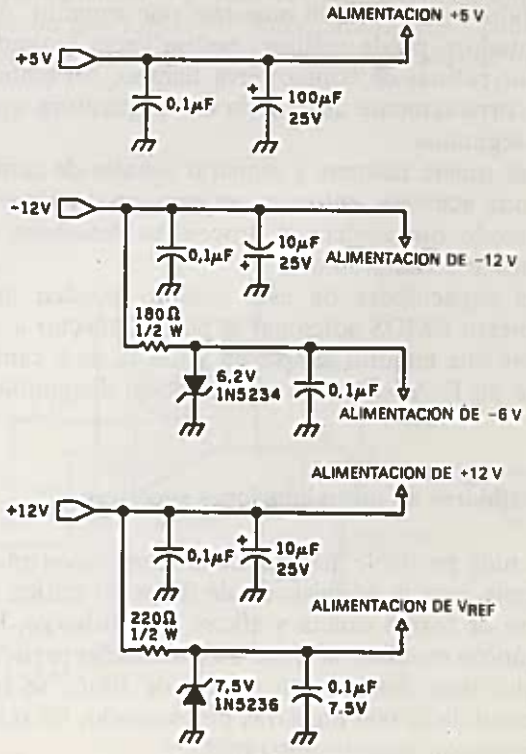


Figura 8.11 *Un diagrama esquemático de un convertidor A/D de aproximaciones sucesivas de 8 bits.*

pero sustituye los contadores por un SAR especial («Succesive Approximation Register», registro de aproximaciones sucesivas). La lógica circular de aproximación sucesiva es explicada mejor en el diagrama de bloques de la figura 8.12.

Inicialmente, la salida del SAR y D/A con conexión mutua están a un nivel «0». Después de un pulso de conversión de arranque, el SAR activa los bits del D/A uno cada vez, comenzando por el MSB. A medida que se activa cada bit, el comparador da una salida indicando que la señal de entrada es más grande, o más pequeña, en amplitud que la salida del D/A. Si la salida D/A es más grande que la señal de entrada, un «0» se pone en este bit particular. Si es menor que la señal de entrada, pondrá ese bit en «1». El registro se desplaza sucesivamente hacia el siguiente bit menos significativo (manteniendo el ajuste en los bits previamente probados) y realiza la misma prueba. Una vez que todos los bits del D/A se hayan probado, el ciclo de conversión está completo. En oposición a los 256 pulsos de reloj del método del contador binario, el período de conversión completo dura solamente 8 ciclos de reloj. Otra conversión comenzaría en el siguiente ciclo de reloj, cuando estuviera en el modo de ejecución libre. Para mantener el valor de 8 bits entre conversiones, ha sido agregado un registro de almacenamiento CI 3 de 8 bits. Para utilizar este A/D, simplemente conecte la salida de este enclavador («latch») a un port de entrada de 8 bits.

Los componentes del circuito de D/A se han cambiado ligeramente con respecto a implementaciones anteriores para aumentar la velocidad y se utiliza un comparador más rápido. Con una frecuencia de reloj de 800 000 Hz, el circuito hará 100 000 conversiones en un segundo. Como son cargadas automáticamente en un registro CI 3 de retención de 8 bits, la actualización es transparente para la computadora y puede ser leída a cualquier velocidad. La frecuencia de muestreo es una función de la frecuencia del reloj. En caso de no ser necesario tener una tan alta frecuencia de muestreo, se puede reducir incrementando el valor de C1. Los convertidores A/D de alta velocidad son susceptibles de selección de componentes y disposición general. Aunque 200 000 muestras por segundo son realizables, 20 000 muestras por segundo podría ser más práctico.

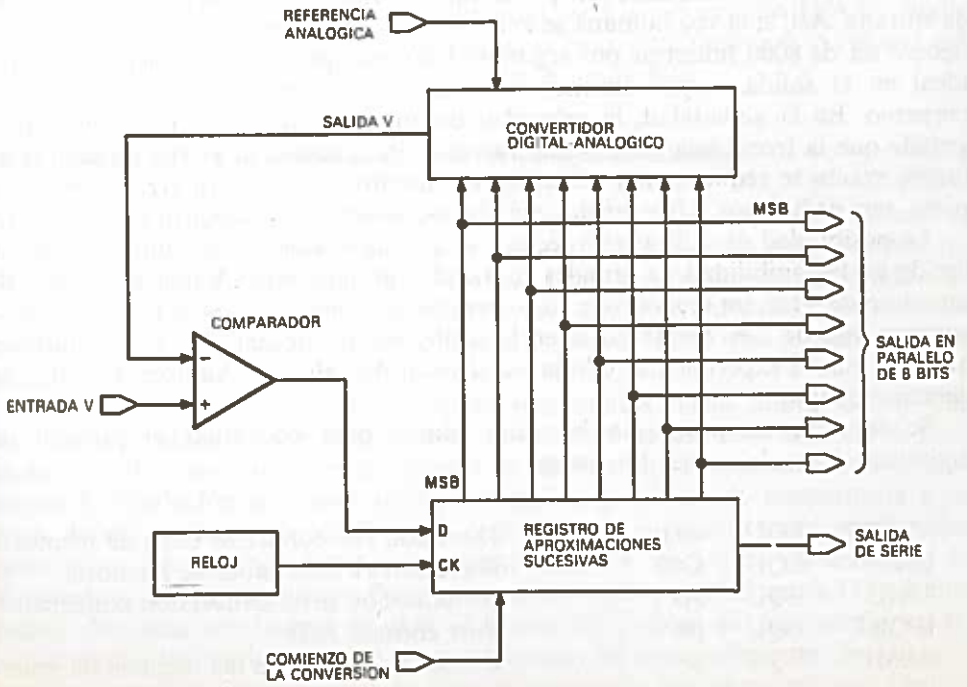


Figura 8.12 *Un diagrama de bloques de un típico sistema de conversión A/D de aproximaciones sucesivas de 8 bits.*

Una singular aplicación para un A/D rápido

Cuando consideremos por primera vez, agregar un A/D al ZAP, nuestros pensamientos se concentraron en controlar algún proceso o en hacer del ZAP un controlador inteligente. En la mayoría de los casos, ello requiere uno de los más simples convertidores A/D que se han examinado. Sin embargo, con la adición de un periférico A/D de alta velocidad, se conciben algunos experimentos más.

Con mucha frecuencia, cuando pensamos en fenómenos analógicos de alta velocidad, sentimos la necesidad de capturar vídeo u otros fenómenos de ancho de banda de alta frecuencia que tengan un nivel de tensión dentro del rango del A/D. Por supuesto, las frecuencias de audio, aunque mucho más bajas que el vídeo, pueden necesitar un A/D de alta prestación para su adecuada representación.

El ancho de banda de la voz humana es de unos 4000 Hz. Estas señales analógicas, cuando se producen en un micrófono y se aplican a un A/D, se pueden convertir en dígitos como cualquier otra forma de onda. Y, si las muestras de nuestra voz son tomadas con la suficiente rapidez y se almacenan los datos acumulados, pueden ser utilizados para reconstruir la misma voz. Esta voz reconstruida se denomina señal vocal digitalizada.

En esencia, la señal vocal digitalizada es simplemente el resultado de una técnica de adquisición de datos normal. Cuando habla ante un micrófono y un amplificador, la voz produce una forma de ondas fluctuantes, cuyo régimen de frecuencia es variable. Si esta señal se aplica a la entrada de un A/D de alta velocidad, y las conversiones se almacenan en memoria, el computador no podría identificar si la fuente fue una voz o la reacción nuclear. Las fluctuaciones analógicas serían digitalizadas a intervalos discretos de muestreo y luego almacenadas. Si las muestras almacenadas se sacan a un D/A a la misma velocidad que fueron tomadas, la voz será reproducida. La fidelidad de esta reconversión es función de la velocidad de muestreo.

La mayor parte del contenido de información o inteligencia en la voz humana se produce en la región de frecuencias por debajo de los 1500 Hz. Obviamente, el muestreo de esta forma de onda a 25 muestras por segundo sería inútil. Debe muestrearse muy rápidamente para retener algo con significación.

Hay una ley específica conocida como «criterio de Nyquist» que se utiliza para determinar la velocidad óptima del muestreo. En teoría, esta ley establece que la frecuencia de la muestra debe ser, por lo menos, dos veces la frecuencia de la onda de entrada. Así, si la voz humana se extiende a 4 Hz, entonces la velocidad mínima deberá ser de 8000 muestras por segundo. Ello presupone, naturalmente, un filtro ideal en la salida, cuya existencia sea tan efímera como la del movimiento perpetuo. En la actualidad, la velocidad del muestreo debe ser 3 ó 4 veces más grande que la frecuencia de entrada más alta. Para convertir la voz en dígitos de forma exacta se requiere una velocidad de muestreo de 12 a 16 Hz. Si, por otra parte, nos dedicamos a frecuencias más bajas, podemos conseguirlo con 3 ó 4 Hz.

La posibilidad de utilizar esta técnica de señales vocales ha de considerarse a la luz de la disponibilidad de grandes cantidades de memoria. A una frecuencia de muestreo de 4 Hz, un segundo de voz corresponde a 4000 octetos de memoria. Si se agregan más de 2 K de memoria en la configuración original del ZAP, entonces quizá se querrá experimentar con la señal vocal digitalizada. Aun con sólo 2 K se deberá oír algo.

Se necesita un programa bastante simple para coordinar el proceso de digitalización y almacenar los datos:

```
START   EQU   400     Dirección HL comienzo tabla de memoria
END     EQU   C00     Dirección H final tabla de memoria
TRIG    EQU   A8      Introducción nivel conversión comienzo
IPORT   EQU   04      Port entrada A/D
SAMP    EQU   38      Constante de tiempo del régimen de muestreo

INP     IN    IPORT   Lectura valor de entrada A/D
        CP    TRIG    Comparar entrada con nivel de disparo
```

```
        JP      NZ, INP     Poner nuevamente en bucle si está por
                            debajo del nivel de disparo
        LD      HL, START   Cargar dirección comienzo tabla
AGAIN   IN      IPORT       Tomar una muestra
        LD      (HL), A     Almacenar muestra en memoria
        CALL    DELY        Retardo entre muestras
        INC     HL
        LD      A, H
        CP      END         Probar para ver si se está al final de la
                            tabla
        JP      NZ, AGAIN   De no ser así, tomar otra muestra
        HALT

DELY    LD      B, SAMP     Arrancar temporizador de retardo
DCR     DEC     B
        JP      NZ, DCR
        RET
```

Cuando se ejecute el programa, explorará el port de entrada de A/D y comparará la lectura con A8 en hexadecimal (aproximadamente un 65% de la escala completa). Cuando la señal vocal está presente, el nivel de audio es presumible que superará este nivel de disparo. Cuando esto pasa, el programa pone la dirección de la tabla de almacenaje y comienza el volcado de muestras de datos en ella, a una velocidad de unas 4000 muestras por segundo. La velocidad viene determinada por el valor de «SAMP». Cuanto más alto sea el número, tanto más baja será la frecuencia de muestreo. Cuando la tabla esté llena, el programa se detiene y la memoria contendrá una representación digitalizada de lo que fue hablado durante el tiempo de muestra. Para 2 K de memoria solamente se captará medio segundo de voz.

Para oír estos datos almacenados, utilice el programa bosquejado en la sección sobre los convertidores D/A. Ponga los límites que definirán el área de la tabla de memoria y luego escoja una constante de tiempo que produzca una extracción de las muestras a la misma velocidad en la que fueron tomadas. (También es posible crear un sistema de reverberación digital utilizando este hardware, pero para obtener una fidelidad aceptable serán requeridos convertidores de 12 ó 14 bits.)

Como la voz digitalizada es una aplicación especial, el circuito D/A se modificará ligeramente para incluir un filtro de paso bajo. Este mejorará la calidad del sonido. El circuito modificado se muestra en la figura 8.13.

Empleo del ZAP en la adquisición de datos de alta resolución

Hasta este punto nuestra exposición se ha centrado en experimentos con el ZAP. Algunos aspectos de este diseño son de utilidad en aplicaciones no adicionales, pero la mayor parte de ellos están concebidos más como ayuda a la enseñanza que como sustitutos de equipos costosos de control. Sin embargo, es posible añadir más interconexiones especializadas al ZAP que le permitan ser utilizadas de esa manera.

Los convertidores A/D de 8 dígitos presentados hasta ahora tienen una resolución limitada y son dispositivos de un solo canal. Son adecuados para mediciones de temperatura en un sistema de calefacción solar, pero es dudoso que tengan la resolución adecuada para controlar el gradiente de temperatura a lo largo de un conducto de calefacción. Los sensores utilizados para medir tales parámetros necesitarían tener una resolución más alta que los sensores de temperatura del aire ambiente. Para un rango de −20 a 108 °C, un A/D de 8 bits podría dar una resolución de 0,5°. En una aplicación de calefacción solar, considerando las variaciones en el movimiento del aire, cobertura de nubes y modelos generales de clima, ésta es toda la resolución que se necesitaría. Dentro del sistema, sin embargo, hay zonas que exigiría mediciones más estrictas.

Un sistema solar es un ejemplo típico. Después de la instalación, el siguiente paso suele ser investigar cómo aumenta su rendimiento funcional. Nueve veces de

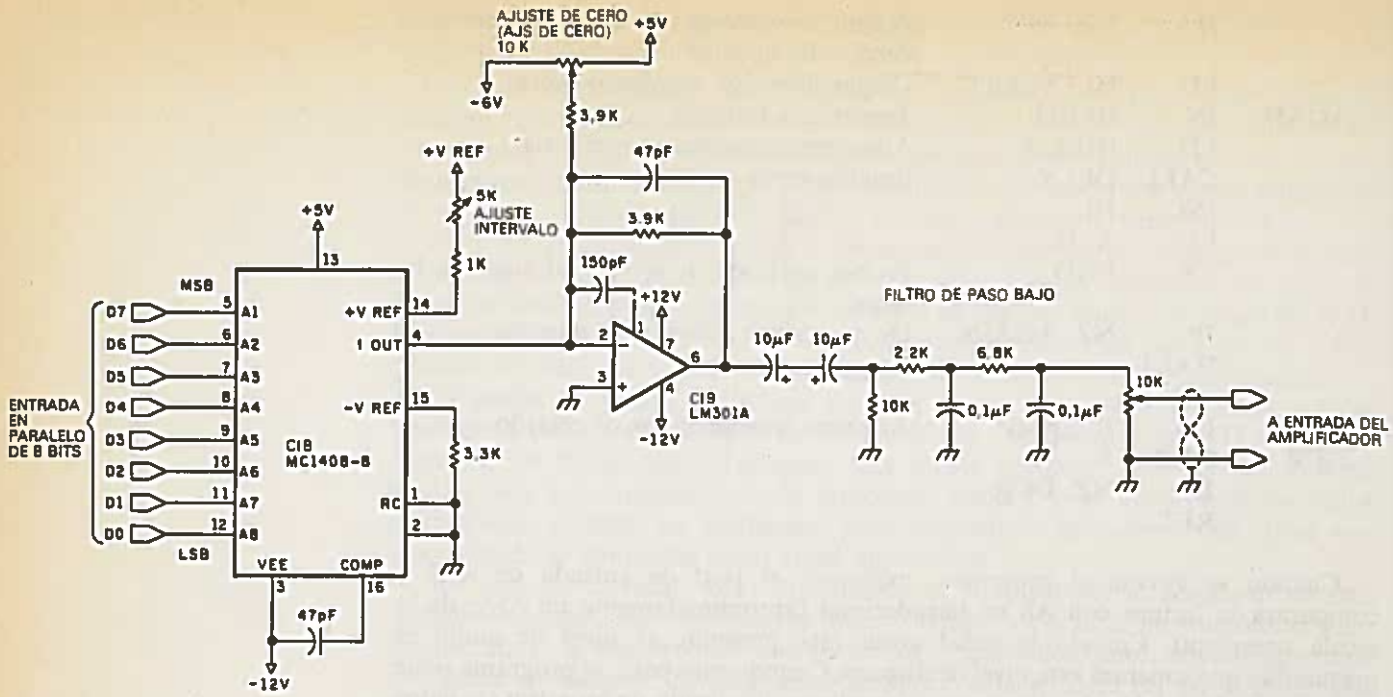


Figura 8.13 *Un convertidor D/A de 8 bits con un filtro de paso bajo.*

cada vez, esto requiere reducir las pérdidas de calor en los tubos y conductos. Una forma de determinar tales pérdidas es poner sensores de temperatura a lo largo del recorrido de distribución del calor y buscar las zonas frías. Las diferencias medidas entre sensores pueden ser muy pequeñas, apenas unas décimas de grado de un orden de magnitud similar, pero las pérdidas totales podrían ser significativas. Medir temperaturas con una exactitud de décimas o de centésimas de grado y mantener el mismo rango dinámico requiere una resolución de más de 8 bits. Se necesita de 10 a 12 bits.

La situación se complica más por el gran número de puntos que se puede necesitar controlar dentro de un sistema. Es raro encontrar solamente un indicador de temperatura en un sistema. Por lo menos, tendrían que haber: aire interior, aire exterior, parte superior del tanque de almacenaje, fondo del tanque de almacenaje, colector y la temperatura del aire de distribución.

Muy pocos sistemas de adquisición de datos utilizan un solo canal. Suelen suministrarse con 8 ó 16 canales multiplexados. La entrada de un convertidor A/D se conmuta (normalmente sobre una base de demanda) entre los canales y los resultados se compilan y promedian mediante la computadora. Esta información puede registrarse en cinta de grabación, transmitirse en serie a otro sistema o utilizarse para la ejecución de una presentación visual en tiempo real. Lo que hace con los datos es una función del programa de aplicación.

Hay diversas formas de configurar ZAP para adquisición de datos de alta resolución. Una consiste simplemente en sustituir el A/D de 8 bits por un convertidor binario de 12 bits. Cuando se ha concluido la conversión, se dispondrá de 12 bits de datos en paralelo. Dependiendo del convertidor elegido, se podría seguir requiriendo muchos componentes analógicos exteriores, pero el proceso es sencillo. Lamentablemente, estos convertidores no son lo que podría llamarse baratos. Aunque su coste se va haciendo cada vez menor, todavía son considerablemente más caros que los convertidores de 8 bits de velocidad similar.

La mayoría de los convertidores binarios de 12 bits son caros porque están concebidos para dar la apariencia de convertidores en paralelo. Al bascular la línea de habilitación de conservación y su código zonal, hay 12 bits de respuesta. Cuando la computadora requiere estos datos, explora, manipula y los almacena en una tabla para su empleo por otros programas. El hacer la sección del hardware de un interfaz de A/D menos cara lleva consigo hacer menos en paralelo. Adoptar el

método alternativo en serie suele requerir más tiempo y manipulación adicional de los datos. Podemos optar por los más bajos gastos y dejar que nuestra computadora haga la mayor parte de trabajo. Ya hemos visto cómo eliminar contadores y lógica de temporización realizando estas funciones en software.

Un interfaz de c.a/c.c de $3^1/_2$ dígitos y 8 canales para ZAP

La solución para la cuestión de alta resolución frente a los gastos viene dada en la forma de un chip de convertidor A/D multiplexado de $3^1/_2$ dígitos. El circuito integrado CMOS MC14433 está concebido primariamente para su empleo en voltímetros digitales, pero, debido a su versatilidad, puede utilizarse en multitud de otras aplicaciones. Se trata de un convertidor de 11 bits de un solo canal, pero se denomina de $3^1/_2$ dígitos. La salida es BCD (decimal codificado en binario) y cubre específicamente un rango de -1999 a $+1999$ en su conteo. Las especificaciones básicas del chip son las siguientes:

Convertidor A/D de $3^1/_2$ dígitos MC14433

Exactitud: $\pm$ 0,05 % de lectura $\pm$ 1 cuenta.
Dos rangos de tensión: 1,999 V y 199,9 mV.
25 conversiones por segundo.
Impedancia de entrada de 1000 megohmios.
Ajuste automático del cero.
Ajuste automático de la polaridad.
Disponibilidad de señales de rango por exceso, por defecto y automático.

El MC14433 es un convertidor A/D integrador de doble rampa modificado y está representado en la figura 8.14. La secuencia de conversión está dividida en dos períodos de integración: desconocido y de referencia. Durante la secuencia de integración de V_{IN} (entrada desconocida), la tensión desconocida se aplica a un integrador con una constante de tiempo de integración definida para un límite de tiempo predeterminado. La salida de tensión del integrador se hace, pues, una función de la entrada desconocida. Cuanto más positiva es la entrada, tanto más alta es la salida del integrador.

Durante el segundo ciclo de la secuencia de integración, una señal de referencia de 2000 V está conectada a V_{IN}. Ello hace que el integrador se desplace hacia cero, mientras que la circuitería digital del chip mantiene el registro del tiempo que lleva alcanzar el cero. La diferencia de tiempo entre las dos secuencias de integración es entonces una función de su diferencia de tensión. Si 2000 V fuera la V_{IN} aplicada, entonces t_2 sería igual a t_1. La tensión desconocida es equivalente a la relación de los períodos multiplicada por la tensión de referencia (V_{REF}). También se conoce como un convertidor medidor de relaciones. La escala completa del convertidor viene determinada por V_{REF}. El cambio de V_{REF} a 0,200 V hará que la salida de conteo de 1999 represente 199,9 mV en lugar de 1,999 V de escala completa.

La salida del chip del voltímetro digital es una combinación de datos en serie y en paralelo. Hay cuatro salidas de selección de dígitos y cuatro líneas de datos en BCD:

Líneas de salidas BCD

Terminal 23	Q3 (MSB)
Terminal 22	Q2
Terminal 21	Q1
Terminal 20	Q0

Salidas de selección de dígitos

Terminal 19	DS1 (MSD)
Terminal 18	DS2
Terminal 17	DS1
Terminal 16	DS0

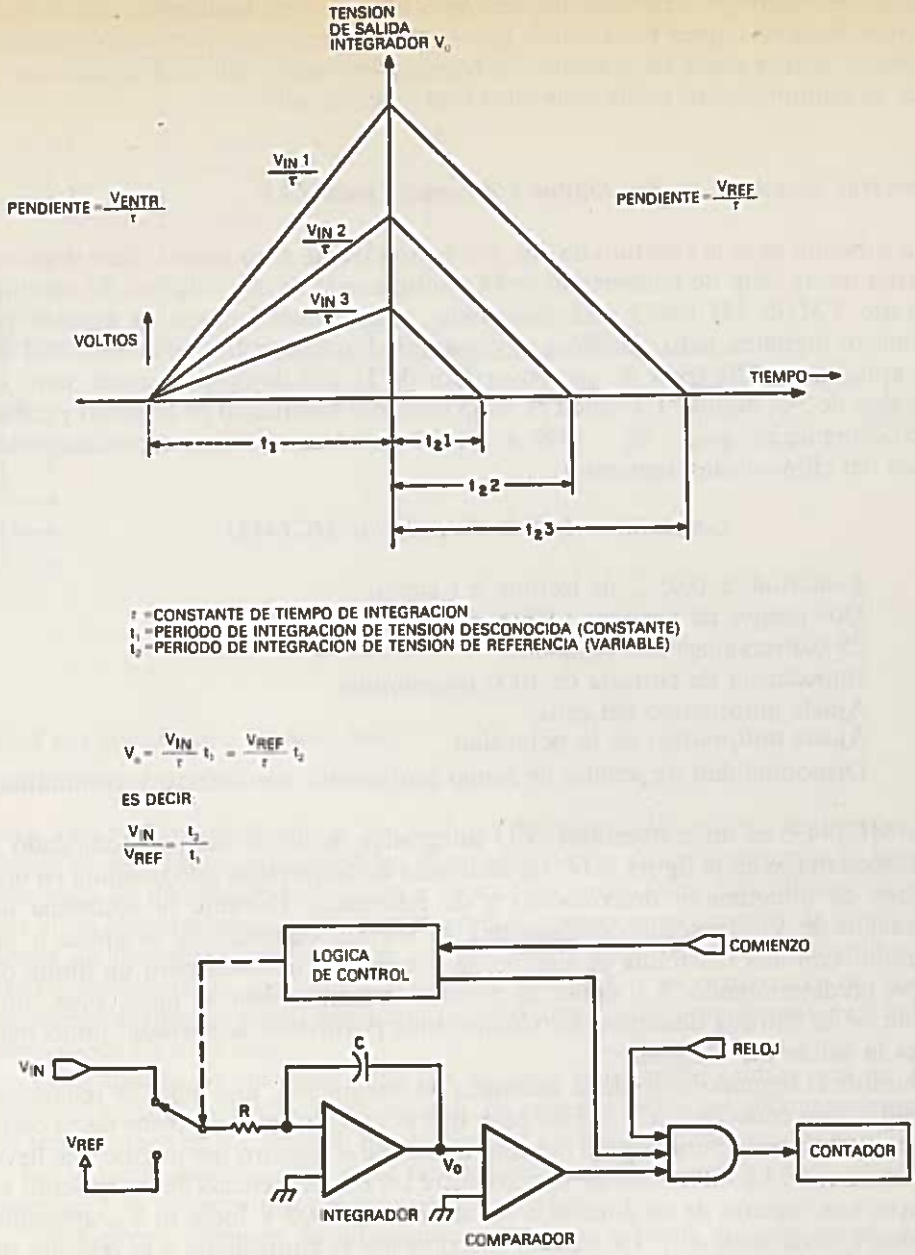


Figura 8.14 *Una representación simplificada de un convertidor A/D de doble rampa.*

Con respecto a lo que la computadora ve a través de las memorias intermedias de salida del 74LS04, la salida de selección de dígitos está a nivel bajo cuando se selecciona el dígito respectivo. El dígito más significativo (1/2 DS1) pasa a nivel bajo inmediatamente después de un pulso de EOC (final de conversión) y va seguido por los dígitos restantes en una secuencia desde MSD. La frecuencia del reloj de multiplexado es la del reloj del sistema dividida por 80; dos períodos de reloj son intercalados entre las salidas de dígitos.

Durante DS1, la polaridad y algunos bits de estado están disponibles. La polaridad está en Q2 y el valor 1/2 dígitos está en Q3. Si Q2 es un «1», entonces la tensión de entrada es negativa, y si Q3 es «0», entonces el 1/2 dígito es un 0.

La figura 8.15 detalla el esquema de la placa de interfaz de 8 canales. Como se muestra, tiene las siguientes capacidades:

Interfaz de voltímetro digital de $3^1/_2$ dígitos del ZAP

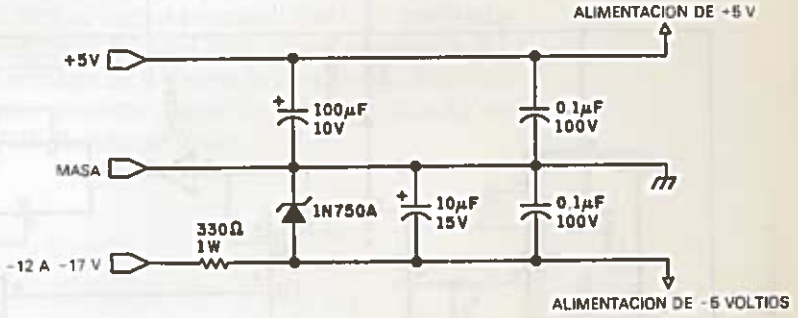
- 8 canales de entrada programables.
- Capacidad de entrada de c.a o de c.c.
- Ganancia programable de 1, 10 ó 100.
- Rangos de 0–200 mV, 0–2 V, 0–20 V o 0–200 V.
- Protección contra sobretensiones de entrada.

CI 1 es el chip de voltímetro digital MC14433. Está ajustado para unas 25 conversiones en un segundo y todas las salidas están provistas de memorias intermedias. CI 2 es un chip de referencia de tensión de precisión que suministra la señal V_{REF}. Es nominalmente 2,5 V y es ajustado a 2,000 V y 0,200 V con dos potenciómetros. Aunque un diodo zener podría proporcionar la misma tensión, la desviación de temperatura asociada con tales componentes los hace inapropiados para esta aplicación.

CI 5 está configurado como un flip-flop SR. Una vez concluida la conversión, una señal EOC activa CI 5, indicando a la computadora que los datos están disponibles. Cuando la computadora termina de leer los datos, reinicializa a este flip-flop y espera la siguiente conversión.

Los CI 1, 2, 3 y 4 constituyen un convertidor de canal único de $3^1/_2$ dígitos. Tienen un rango o de 0,200 V o de 2,000 V determinados por V_{REF}. Para lograr la operación multicanal y la capacidad de c.a es necesario poner un multiplexor de entrada y un convertidor de c.a/c.c en frente de CI 1.

La figura 8.16 muestra el circuito de la referencia de tensión y de selección de rangos en este interfaz. El MC14433 puede cubrir 0–199,9 mV o 0–1,999 V. Los rangos dependen del nivel de V_{REF}. Cuando B5 del port está a nivel bajo, los conmutadores 5 y 6 están en la posición mostrada. Esto aplicaría 2,000 V a la entrada de V_{REF} y ajustaría la constante de tiempo de integración con una resistencia de 82 kΩ. Con B5=0, V_{REF} es 0,200 V y la resistencia de integración es de los 10 k Ω.



1. TODAS LAS RESISTENCIAS SON DE 1/4 W 5 %, A NO SER QUE SE INDIQUE LO CONTRARIO
2. TODOS LOS CONDENSADORES SON DE CERAMICA DE 100 V, A NO SER QUE SE INDIQUE OTRA COSA

CI #	TIPO	+5V	-5V	MASA
1	MC14433	24	12	13
2	MC1403	1		3
3,4	74LS04	14		7
5	7474	14		7
6,7	CD4053	16	7	8
8	CD4051	16	7	8
9	7445	16		8
10	LM324	4	11	

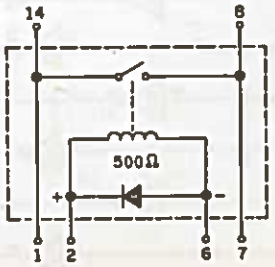
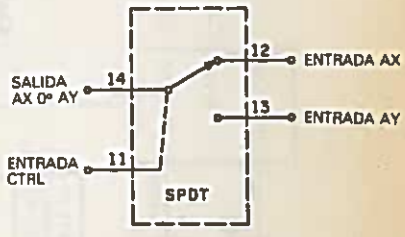


DIAGRAMA DE TERMINALES DEL RELE SIGMA TIPO 1911TE1A2 55, CAPSULA DIP DE 14 TERMINALES

CTRL	SALIDA
0	AX
1	AY



DESCRIPCION FUNCIONAL DE UNA SECCION DE CONMUTACION UNICA (1 DE 3) DE UN CONMUTADOR CMOS CD4053

Figura 8.15 *Un interfaz de voltímetro digital de 8 canales de $3^1/_2$ dígitos de 0–200 V c.a/c.c* *(continúa en la página siguiente).*

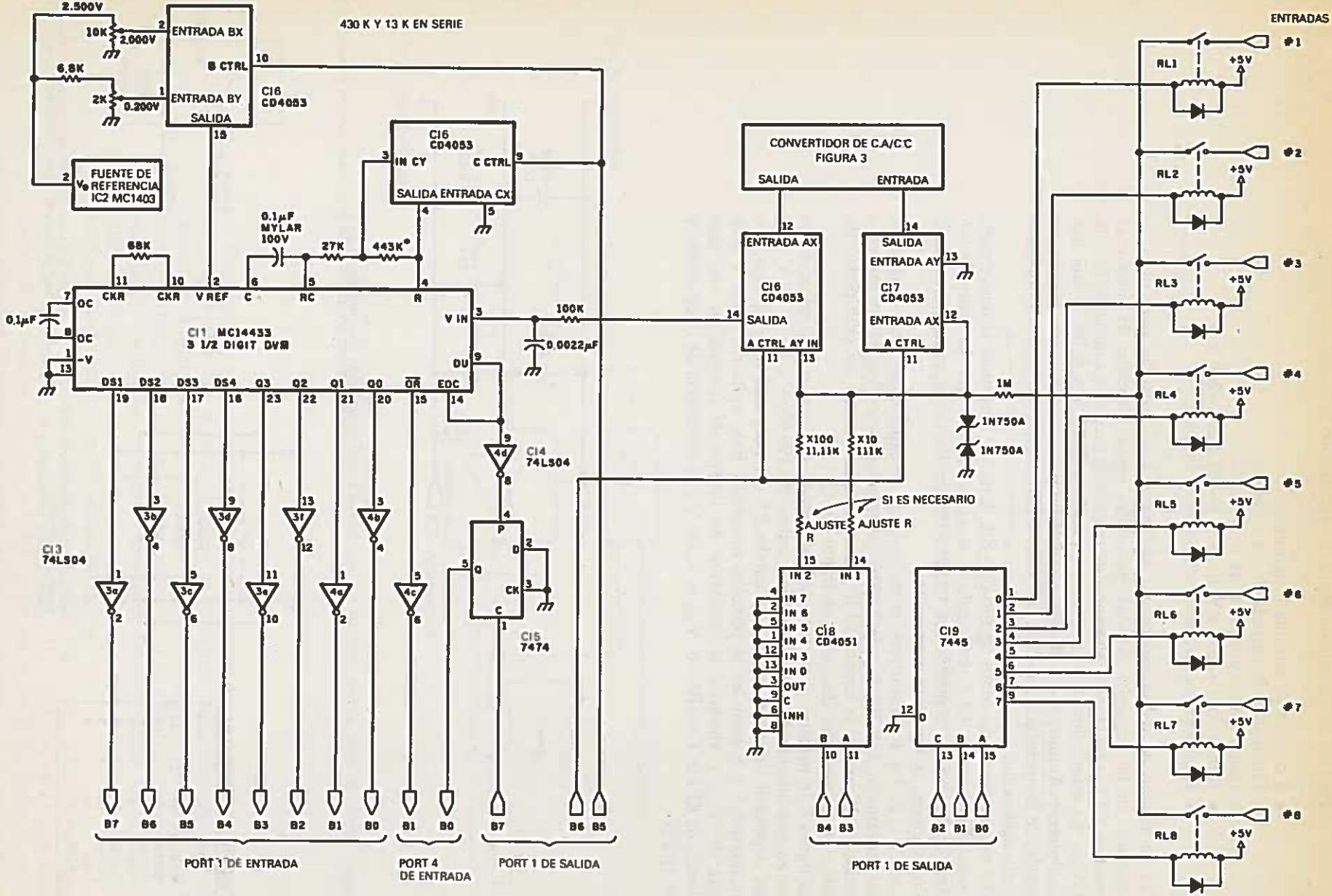


Figura 8.15 *(Continuación.)*

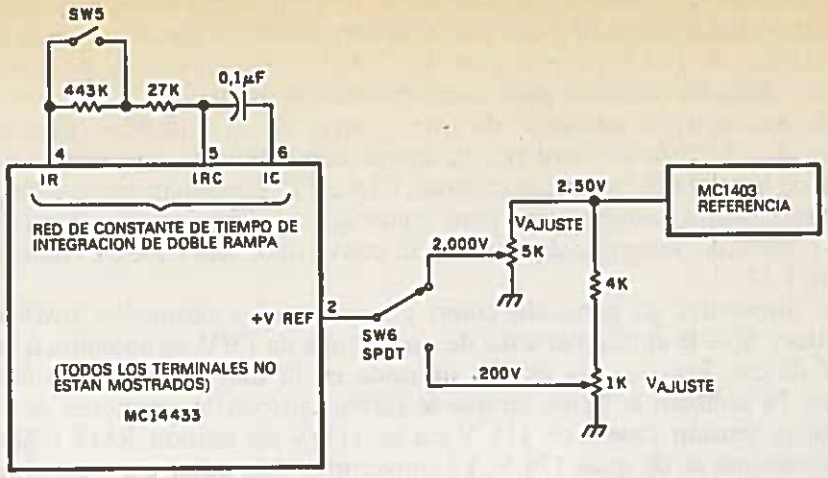


Figura 8.16 *Circuitos de modificación de constante de tiempo de integración y referencia de tensión para el voltímetro digital.*

La figura 8.17 ilustra el subsistema de entrada en términos simplificados. SW1 y SW2 representan la sección de selección de ganancia. Como se muestra, la ganancia es 1 y no se activa ninguna red divisora. Cuando se cierra un relé de entrada (controlado mediante CI 9), la tensión de entrada de ese canal se envía directamente a la entrada de CI 1, a través de una resistencia de 1 MΩ. Si el interfaz se establece para c.c y para una ganancia de 1, una señal de entrada de 1,400 V sería leída directamente en el canal 3 como 1,400 V por el chip de voltímetro digital. Si, no obstante, fueran aplicados bruscamente 150 V, sería derivado a través de Z1 y Z2, que protegería a CI 1. Los datos leídos por la computadora indicarán una condición de fuera de rango, porque la entrada estaría derivada a 4 V.

Cerrando SW1 o SW2 se forma una red divisora que permite a la computadora leer estas tensiones más altas. Un divisor 10:1 se forma cerrando SW1. El resultado es una red divisora constituida por la resistencia R1 de 1 MΩ y una resistencia R2 puesta a tierra de 111 kΩ. Una señal de entrada de 8 V sería leída como 0,800 V en la entrada CI 1. El programador debe tener presente que el divisor fue utilizado en ese canal y multiplicar la respuesta por 10 cuando se grabe.

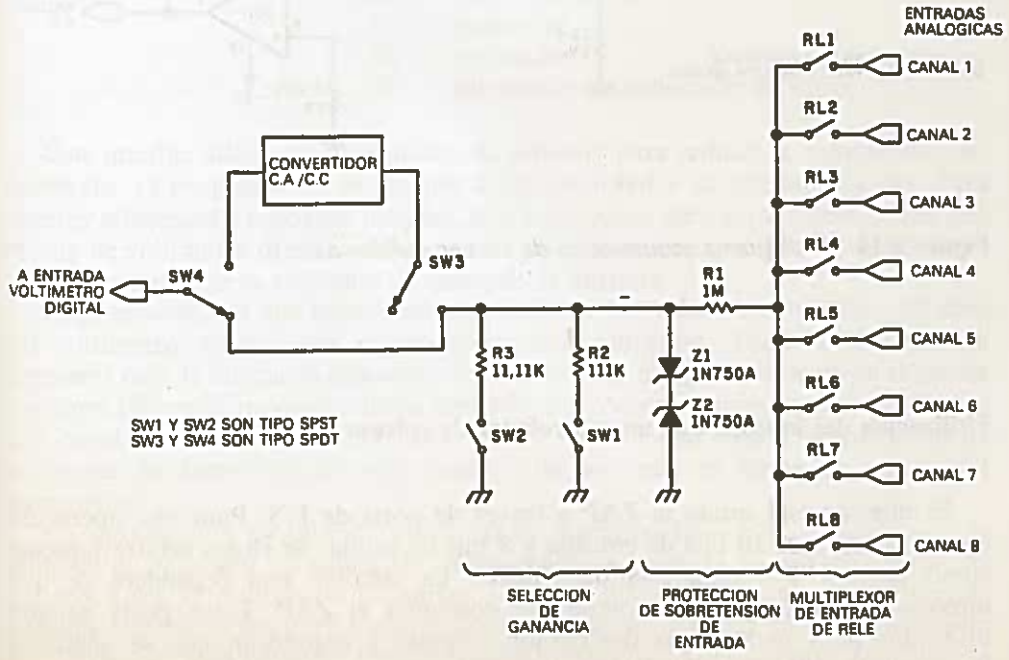


Figura 8.17 *Secciones de acondicionamiento de entrada de voltímetro digital (DVM).*

Cerrando SW2 se forma un divisor 100:1. La matemática es la misma salvo que la resistencia R3 es ahora de 11,11 kΩ. Una entrada de 8 V pasaría a ser de 0,080 V y una entrada de 150 V pasaría a ser de 1,500 V. Obviamente, la selección apropiada del rango es necesaria para hacer máxima la resolución.

Una característica adicional de este interfaz es la capacidad para admitir entradas de c.a. Esto se logra por la simple conversión de una señal c.a en c.c después de la salida de la sección divisora. CI 6 e CI 7 funcionan como conmutadores bidireccionales monopolares para controlar mediante puerta el convertidor dentro o fuera del recorrido de la señal. El convertidor real c.a/c.c es mostrado en la figura 8.18.

Este dispositivo es conocido como un convertidor promedial RMS (media cuadrática). Si se le aplica una señal de c.a máxima de 1,0 V, se obtendrá a la salida 0,707 V de c.c. Esta es una técnica utilizada en la mayoría de los multímetros digitales. Es también la forma en que se suelen expresar las tensiones de c.a. Por ejemplo, la tensión casera de 115 V c.a es 115 V de tensión RMS (eficaz). La tensión máxima es de unos 176 V. El convertidor deja pasar c.a y c.c porque no hay un condensador de bloqueo a la entrada. Si, inadvertidamente, se conmuta una señal de c.c, multiplicará la lectura por 1,414 ($\sqrt{2}$).

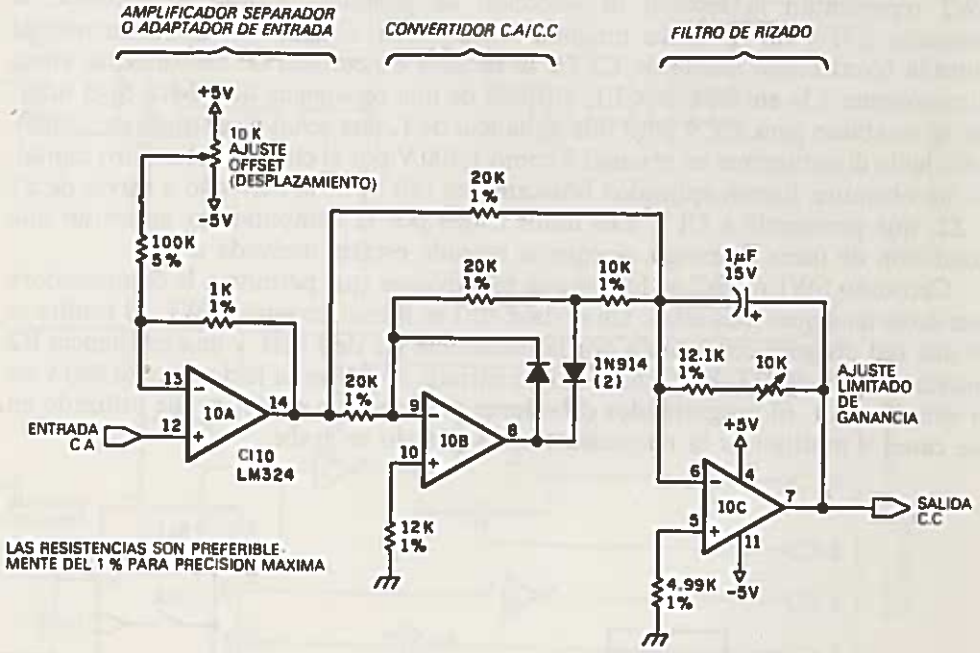


Figura 8.18 *Un diagrama esquemático de un convertidor c.a/c.c.*

Utilización del interfaz con un controlador de software

El interfaz está unido al ZAP a través de ports de E/S. Para una operación completa requiere 10 bits de entrada y 8 bits de salida. Se eligen arbitrariamente como ports 1 y 4 para esta descripción. La elección real dependerá de qué direcciones estén cableadas cuando se configura el ZAP. Estos ports no son utilizados para nada en la descripción original y requerirán que se añada el hardware de port adecuado. En resumen, los requerimientos de E/S para el interfaz DVM (voltímetro digital) son los siguientes:

Octeto de salida de orden (port 1 salida)

B7	habilitación o inhibición de EOC	inhibición=1; habilitación=0
B6	selección de c.a o c.c	c.a=0; c.c=1
B5	rango de 2,0 V o de 0,2 V	2,0 V=0; 0,2 V=1
B4 B5	código de ganancia	0,0=X1 0,1=X10 1,0=X100
B2 B1 B0	código de canal	canales 0–7 binario

Octeto de entrada de estado (port 4 entrada)

B7 B6 B5 B4 B3 B2	no usado	
B1	fuera de rango	
B0	final de conversión	

Octeto de entrada de datos (port 1 entrada)

B7	1.^{er} dígito	
B6	2.° dígito	habilitación dígito
B5	3.^{er} dígito	
B4	4.° dígito	
B3 B2 B1 B0	valor en BCD	

Cuando B7=0, entonces:

B6	
B5	no usado
B4	
B3	valor 1/2 dígito
B2	polaridad
B1	no usado
B0	bit estado autoselección de rango

Este interfaz utiliza un controlador de software para reducir la complejidad del hardware. El programa no es distinto a un controlador de comunicaciones. Para obtener eficazmente datos del interfaz, la computadora debe estar sincronizada con el chip de voltímetro digital y debe realizar una secuencia específica de operaciones para demultiplexar la corriente de datos de la entrada.

El programa real que interconecta y almacena los valores procedentes del chip del voltímetro digital está escrito como una subrutina. Toda la información necesaria para la adecuada ejecución del controlador es proporcionada en el par de registros DE en el momento de la llamada. Su contenido comunicará al interfaz qué canal ha de establecerse, si debe ser c.a o c.c y qué V_{REF} y ganancia ha de utilizarse. Se convierte un solo canal cada vez que se llama la rutina del controlador.

La información establecida en el par de registros DE en el momento de la llamada es el octeto de salida de orden (entrada, port 1), y cada bit tiene las designaciones anteriormente indicadas. La única diferencia es que el bit 7 (bit de habilitación/inhibición para el convertidor A/D) se envía como un nivel lógico «0» cuando se hace una llamada. El controlador lo pondrá a «1» para una condición de habilitación después de que se haya puesto en trabajo el relé adecuado y permitido un retardo de rebote de 1,3 ms.

La demultiplexión de la salida del chip del voltímetro digital es bastante sencilla. Después de la llamada, las salidas al interfaz cierran los conmutadores apropiados y el procesador central se mantiene en un bucle a la espera de una señal de final de la conversión. Cuando esto sucede, el programa sabe que los siguientes 4 dígitos de datos son los que quiere. El chip de DVM (voltímetro digital) activa cada una de las líneas de selección de dígitos de forma sucesiva y el programa registra cada vez los valores de las 4 líneas de datos de BCD. Elimina los bits de estado y polaridad del octeto MSD $^1/_2$ dígito y da un nuevo formato y almacena el valor de entrada de tensión en 4 octetos de memoria. Los tres dígitos enteros son almacenados en notación BCD y ocupan 3 de los octetos. El $^1/_2$ dígito, la polaridad y la indicación de fuera de rango se localiza en el cuarto octeto. La polaridad se indica poniendo a «1» el MSB. Una lectura positiva es un nivel lógico «1» y una entrada negativa es un nivel lótico «0». El valor $^1/_2$ dígito solamente puede ser un «0» y un «1» y ocupa el LSB de la cantidad. La indicación de fuera del rango se controla por una pequeña manipulación de programa. La indicación de fuera del rango se controla por una pequeña manipulación de programa. Si el controlador detecta que la lectura entrante no está dentro del rango, activa el equivalente de +2 en el octeto de $^1/_2$ dígito. Obviamente, ésta es una condición ilegal para un voltímetro digital que solamente es capaz de contar hasta 1999. El programador que utilice estos datos almacenados deberá revisar los límites de estos datos antes de actuar sobre ellos.

Cuando el controlador completa su operación, habrá adquirido una lectura de $3^1/_2$ dígitos y almacenado como 4 octetos en una tabla especial en memoria. Los 8 bits canales de datos constituyen una tabla de 32 octetos. La posición de los datos de un canal particular se encuentra mediante una expresión simple:

Los datos de 4 octetos comienzan en la posición de memoria $L+4(N-1)$

En donde L=dirección de comienzo de tabla de memoria
N=número de canales (1 a 8)

La figura 8.19 es el listado ensamblador del programa que ejercita este interfaz de DVM. Cuando se ensambla, ocupa menos que una página de memoria.

Nota: Se deberá tener precaución cuando se midan señales de c.a con este interfaz. La tierra en el interfaz de DVM es la misma que la de la computadora y existe un corto circuito potencial, a menos que esté aislada la fuente de alimentación de la computadora o la tensión medida.

```
0100 %
0110 %%% CONTROLADOR DE CONVERTIDOR A/D DE 3 1/2 DIGITOS MC14433
0120 %
0128 % REV. 1.9
0130 %
0140 DIP      EQU   1        NUMERO DE PORT DE ENTRADA DE DATOS
0150 SIP      EQU   4        NUMERO DE PORT DE ENTRADA DE ESTADO
0160 COP      EQU   1        NUMERO DE PORT DE SALIDA DE ORDEN
0170 EEOC     EQU   200      HABILITACION DE ENTRADA EOC
0180 DEOC     EQU   000      INHIBICION DE ENTRADA EOC
0190 %
0200 %
0210 % MEMORIAS INTERMEDIAS DE DATOS DE CANALES CONVERTIDOS
0220 %
0230 CHAN0    DW    000000
0240          DW    000000
0250 CHAN1    DW    000000
0260          DW    000000
0270 CHAN2    DW    000000
0280          DW    000000
0290 CHAN3    DW    000000
0300          DW    000000
0310 CHAN4    DW    000000
0320          DW    000000
0330 CHAN5    DW    000000
0340          DW    000000
0350 CHAN6    DW    000000
0360          DW    000000
```

Figura 8.19 *Un listado del programa en lenguaje ensamblador que ejercita el voltímetro digital.*

```
0370 CHAN7   DW    000000
0380         DW    000000
0390 %
0400 % MEMORIAS TEMPORALES DE DATOS INTERMEDIOS
0410 %
0430 CHAN    DB    000          NUMERO DE CANALES ACTUALES
0440 CCP     DW    000000       PARAMETRO DE CANAL DE ORDEN
0460 %
0470 %
0480 %%% ARRANQUE DEL CONVERTIDOR A/D
0490 %
0550 %
0560 START   LD    A,E
0570         LD    (CCP),A
0580         AND   007
0590         LD    (CHAN),A
0600         LD    IX,CHANO
0910         LD    D,0
0920         LD    E,A
0930         SLA   E            CALCULAR DESVIACION AMPLIFICADOR SEPARADOR
0940         SLA   E
0950         ADD   IX,DE
0960 %
0970 % SELECCION DE CANAL E INICIO DE LA CONVERSION
0980 %
0985         LD    B,3          ESTABLECER CONTEO DE CICLO
0990 SCSC    LD    A,(CCP)
1000         OUT   COP          SELECCION DE CANAL
1005         CALL  DELAY
1010         OR    EEOC         HABILITACION DE SALIDA EOC
1020         OUT   COP          ORDEN DE CONVERTIDOR A/D
1030 %
1040 % WAIT  FOR   EOC
1050 %
1060 WEOC    IN    SIP          LECTURA DEL ESTADO DEL CONVERTIDOR
1070         BIT   0,A          PRUEBA PARA EOC
1080         JR    Z,WEOC       SALTO SI NO ESTA LISTO
1085         DJNZ  SCSC
1090         BIT   1,A          PRUEBA PARA SOBRE RANGO
1100         JR    NZ,OVER      SALTO SI ES VERDADERO
1110 %
1120 % CONVERSION HECHA; PROCESAR PRIMER DIGITO (MSD)
1130 %
1140 MSDO    LD    B,200        SELECCIONAR DIGITO 1
1150         CALL  RDIG         ESPERAR Y LEER DIGITO 1
1160         CPL
1170         RRCA  J            U     CAR EL VALOR DEL DIGITO A LA DERECHA
1180         RRCA
1190         RRCA
1200         AND   1            AISLAR
1210         LD    E,0          INICIALIZAR OCTETO DE ESTADO
1220         BIT   2,D          PROBAR POLARIDAD
1230         JR    NZ,MSD3      SALTO SI ES POSITIVO
1240         LD    E,200        CARGAR SIGNO DE POLARIDAD
1440 %
1450 % CONSERVAR MSD Y LA POLARIDAD ACTUAL
1460 %
1470 MSD3    OR    E            AGREGAR SIGNO DE POLARIDAD AL MSD
1480         LD    (IX+0),A     CONSERVAR EN MEMORIA INTERMEDIA DE DATOS
1500 %
1510 % PROCESAR 2.º DIGITO
1520 %
1530         RRC   B            SELECCIONAR DIGITO 2
1540         CALL  RDIG         ESPERAR Y LEER DIGITO
1550         AND   017          AISLAR
1560         LD    (IX+1),A     ALMACENAR SEGUNDO DIGITO
1570 %
1580 % PROCESAR EL 3.º DIGITO
1590 %
1600         RRC   B            SELECCIONAR 3.º DIGITO
1610         CALL  RDIG         ESPERAR Y LEER DIGITO
1620         AND   017          AISLAR
1630         LD    (IX+2),A     ALMACENAR
1640 %
1650 % PROCESAR 4.º DIGITO
1660 %
1670         RRC   B            SELECCIONAR 4.º DIGITO
1680         CALL  RDIG         ESPERAR Y LEER DIGITO
```

Figura 8.19 *Continuación.*

```
1690            AND   017          AISLAR
1700            LD    (IX+3),A     STORE
1710 RAPUP      RET
1720 %
1730 % VALOR DE SOBRERANGO EN MEMORIA INTERMEDIA DE DATOS
1740 %
1750 OVER       LD    A,2          CARGAR VALOR MSD
1760            LD    (IX+0),A
1770            XOR   A
1780            LD    (IX+1),A     CARGAR VALORES LSD
1790            LD    (IX+2),A
1800            LD    (IX+3),A
1810            LR    RAPUP
1870 %
1880 %
1890 % LEER RUTINA DE DIGITOS
1900 %
1910 RDIG       IN    DIP          CONVERTIR A LOGICA VERDADERO ALTO
1930            LD    D,A          CONSERVAR LA COPIA
1940            AND   B            PRUEBA PARA DIGITO DADO LISTO
1950            JR    Z,RDIG       SALTO SI NO LO ESTA
1960            LD    A,D          RESTAURAR UN REGISTRO
1970            RET                RETORNAR A SOLICITAR (REALIZADOR
                                     DE LLAMADA)
1980 DELAY      LD    C,377
1990 DEL1       C
2000            RET   Z
2010            JR    DEL1

KEYIN1      %MEM (DIRECCION) NEXT, (DATO) NEXT... (DATO) EXEC
CONTROLADOR PARA LA SALIDA DEL TELETIPO
```

Figura 8.19 *Continuación.*

Aplicaciones potenciales

A juicio del autor, la adquisición de datos es una aplicación natural para el ZAP. El interfaz anteriormente descrito puede ser utilizado en un sistema de calefacción solar para controlar y registrar datos pertinentes. Utilizando los medios que provee el monitor ZAP y la rutina de interfaz del DVM, un tabulador-registrador de datos de 8 canales resulta práctico. En general, todo lo que se requeriría es un programa de supervisor que llame al DVM 8 veces para obtener las 8 entradas del sensor. Entonces establece los límites de la tabla de memoria para una subrutina de salida en serie y almacena las lecturas en una casete. Esto podría realizarse de forma continua o a intervalos regulares. El sistema definitivo incluirá un reloj de tiempo real, de tal forma que estas lecturas pudieran registrarse, así como los tiempos en que fueron efectuadas.

Reloj de tiempo real

Si el ZAP va a ser utilizado para funciones de control o de adquisición de datos críticos, debe prestarse consideraciones a la sincronización en tiempo real con los eventos del proceso. Una definición simple de un sistema de tiempo real es una que responda a la necesidad de acción en un período de tiempo proporcional a la urgencia de la necesidad. Se reduce al hecho de que la computadora debe ser capaz de realizar una acción específica en un tiempo específico. Para que esto suceda, la computadora debe ser capaz de «marcar la hora».

Podemos realizar esto utilizando aplicaciones de software o aplicaciones de hardware. La técnica más simple es utilizar un circuito de reloj (figura 8.20) para proporcionar un «tic-tac» de tiempo a la línea de interrupción no enmascarable del procesador central. Puede ser cada 1/60, 1/10 ó 1 segundo, como se indica en el

esquema. Cuando la computadora acusa recibo de la interrupción, primero conserva todos los registros del programa que estaba ejecutando y luego atiende a la interrupción de tiempo real. Frecuentemente, la primera acción es incrementar un contador interno que mantiene el registro del tiempo transcurrido. Suele ser un valor equivalente al número total de «tic-tac» reloj, ya sea en segundos o en milisegundos. Una vez que ha sido establecido este intervalo periódico, es fácil para la computadora realizar funciones de tiempo real.

Las resoluciones de reloj hasta milisegundos suenan bien y hacen la temporización de los intervalos de gran precisión. Sin embargo, se duda de que la mayoría de los constructores del ZAP quisieran utilizar dicho interfaz habida cuenta del complejo software que implica. Se suele preferir un interfaz que sea más fácil de implementar y de utilización más probable.

Esencialmente, el tipo de sistema de tiempo real más atractivo para los propietarios del ZAP tiene una resolución de quizá 1 minuto mucho mejor que 1 ms. También es mejor si puede ser leído directamente en horas y minutos en lugar de leerlo como si fuera un conteo de reloj total. Un beneficio directo es el de la reducción de los gastos generales. La computadora no tiene que confirmar la actualización del reloj ni explorar los indicadores de estado como es usual. A primera vista, podría no parecer que hay mucho ahorro, pero algunas rutinas pueden utilizar hasta el 10% del tiempo del procesador en manipular una interrupción de reloj de milisegundo.

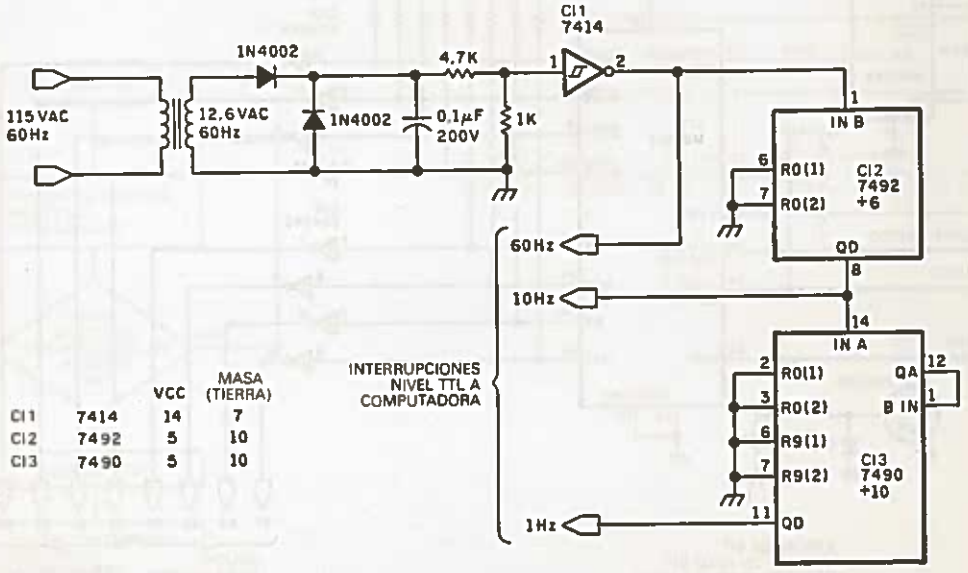


Figura 8.20 *Un generador simple de base de tiempos para un reloj de tiempo real controlado por interrupción.*

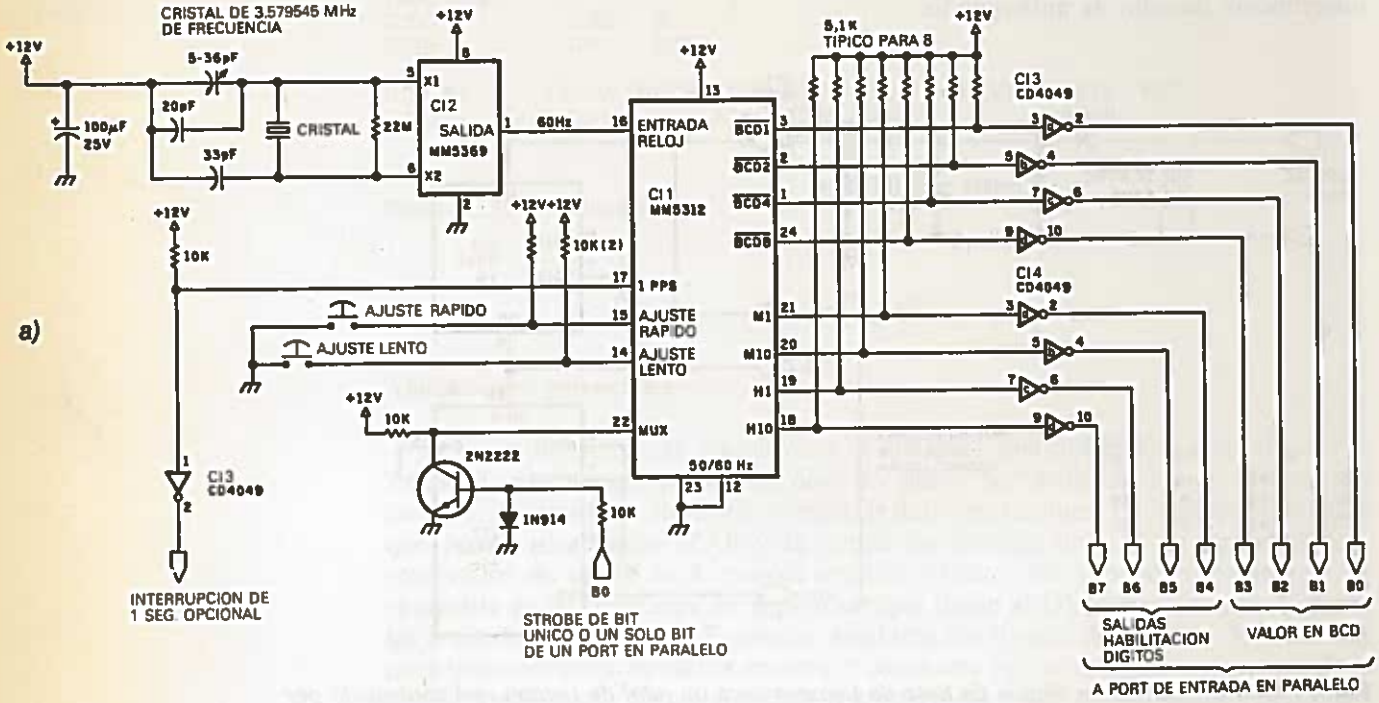
Un viejo chip de reloj al rescate

La forma más fácil de proporcionar una entrada hora a hora y minuto a minuto es interconectando la computadora a un chip de reloj MOS/LSI similar al que se encuentra en la mayoría de los relojes digitales. Hay dos métodos para diseñar un interfaz de reloj. Un método es dejar funcionar el circuito del reloj independientemente de la computadora, unido de tal forma que la computadora pueda controlar las líneas de salida y extraer un valor de tiempo al vuelo. El software necesario para este método sería muy similar al del interfaz de DVM antes descrito. El otro método, que es preferible porque implica menos software, es dar a la computadora el control completo sobre el flujo de información del reloj de una manera síncrona.

La figura 8.21 muestra dicho interface de reloj. Este circuito, preajustado manualmente para mantenerlo simple, es gobernado por la computadora. El circuito básico de 4 chips consiste en un chip de reloj digital de salida de cuatro

dígitos BDD/siete segmentos, MM5312, un generador de base de tiempos MM5369 y dos buffers (amplificador adaptador) MOS a TTL para enviar datos al procesador.

El tiempo se ajusta en el chip poniendo a tierra las líneas de ajuste de tipo lento y rápido, terminales 14 y 15. Para saber lo que se está estableciendo, debe leerse el interfaz al mismo tiempo y visualizar el tiempo en la pantalla de dirección hexadecimal de 4 dígitos que ya es parte del ZAP expandido. El tiempo es leído a partir del interfaz como 4 números decimales codificados en binario (BCD). Las 8 líneas de entrada a la computadora están unidas a un port de entrada en paralelo de 8 bits, y están divididas entre 4 líneas de habilitación de dígitos y cuatro líneas de valor de dígito en BCD. Los datos aparecen como habilitación de dígitos y como un número asociado en BCD. Los datos de decenas de minutos son leídos en B0 a B3 inclusive cuando B5 está a nivel alto (B4, B6 y B7 están a nivel bajo). De manera similar, B0 a B3 retendrán la cantidad de decenas de horas cuando B7 está a nivel alto. La lógica de interfaz permanecerá en un dígito particular hasta que reciba instrucciones para proseguir con el siguiente dígito. El secuenciamiento está bajo el control del programa y utiliza un bit de salida de un port en paralelo adecuado.



CI #	TIPO	+5V	+12V	MASA
1	MM5312		13	23
2	MM5369		8	2
3	CD4049	1		8
4	CD4049	1		8
5	7406	14		7
6	74147	16		8
7	CD4050	1		8

PARA FUNCIONAMIENTO DURANTE 24 HORAS, EL TERMINAL 11 SE PONE A MASA EN EL MM5312

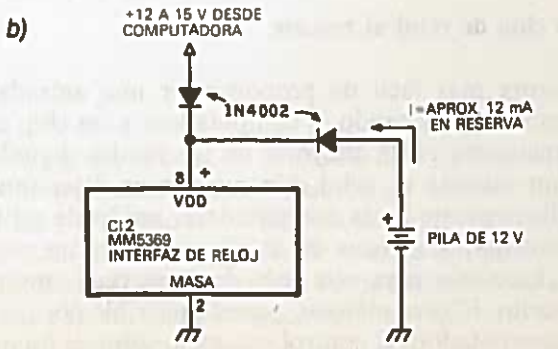


Figura 8.21 *Un diagrama esquemático de un interfaz de reloj de tiempo real.*
a) Utilizando un chip de reloj digital MOS.
b) Con batería de reserva.

La figura 8.22 muestra cómo la línea de multiplexor es controlada en esta aplicación. Un bit de un port de salida es utilizado para pulsar la entrada del multiplexor en el terminal 22. (Todo lo que se requiere es un pulso de 1 ms. Como una alternativa, un monoestable podría ser disparado por una línea de «strobe» decodificado de un port no cableado.) En cualquier momento, una de las cuatro líneas de habilitación de dígitos estará a nivel bajo y un valor de dígito estará en las líneas de salida en BCD. Basta determinar qué dígito es y almacenar el valor. Seguidamente, pulsamos la entrada del multiplexor para habilitar el siguiente dígito y conservarle también. De forma concebible, sólo se requiere cuatro interaciones de este procedimiento para obtener una lectura completa de cuatro dígitos. Si se prefiere un sistema más ordenado, se puede seguir el flujo de programa ilustrado en la figura 8.23. La única diferencia es que espera hasta el principio de los ciclos de chip antes de almacenar las lecturas.

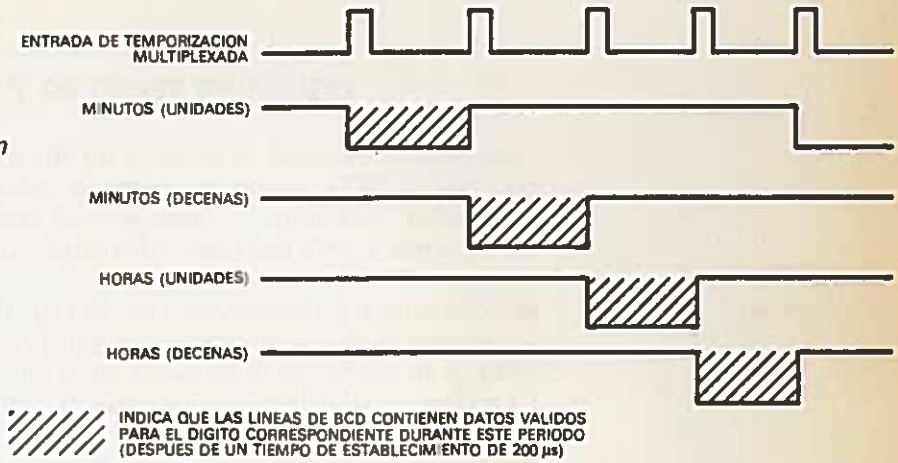
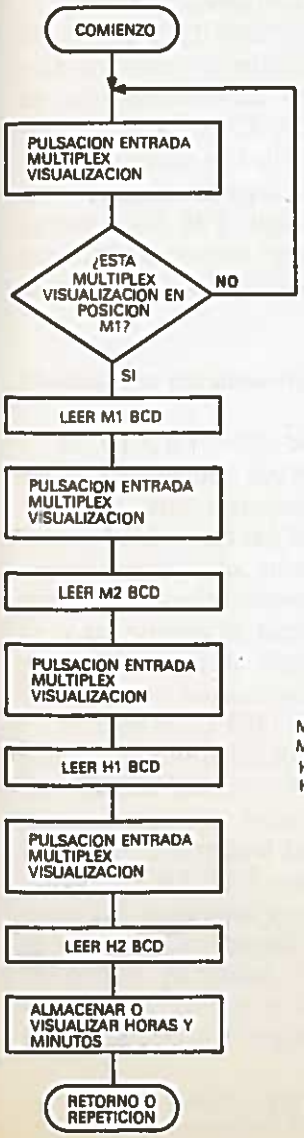
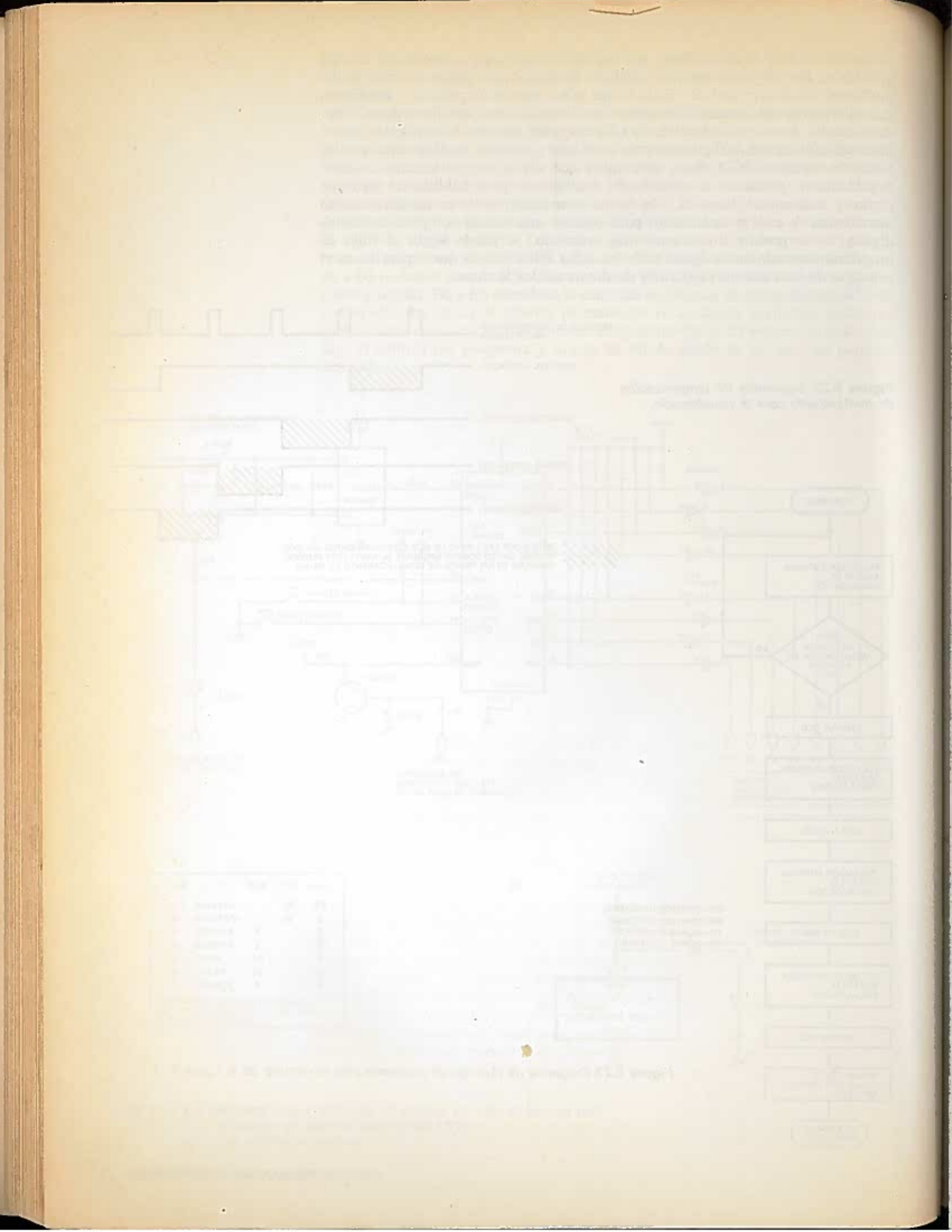


Figura 8.22 *Secuencia de temporización de multiplexado para la visualización.*



M1 = MINUTOS (UNIDADES)
M2 = MINUTOS (DECENAS)
H1 = HORAS (UNIDADES)
H2 = HORAS (DECENAS)

Figura 8.23 *Diagrama de flujo de un programa para el circuito de la figura 8.21.*



Capítulo 9

CONSTRUYA UN TERMINAL DE TRC

TERMINAL VERSATIL CON TRC Y DE COSTE REDUCIDO

En este capítulo se describe el diseño de un terminal de buenas características, coste reducido y provisto de un tubo de rayos catódicos (TRC). Los dos dispositivos MOS/LSI de la Standard Microsystems Corporation reducen el número de elementos necesarios para construir el terminal con TRC y aun mejoran sus posibilidades.

Los dos dispositivos, el CRT 5027, que es un temporizador y controlador de vídeo, y el CRT 8002*, que es un controlador de las propiedades de la representación en vídeo, constituyen prácticamente la totalidad de la circuitería de la parte de representación del terminal con TRC. (Véanse las características de ambos CI en los Apéndices C8 y C9.)

El terminal está diseñado para funcionamiento autónomo y para comunicarse con cualquier computadora a través del interfaz RS-232C. Si la representación hexadecimal de 6 dígitos resultase insuficiente en el ZAP ampliado, entonces el experimentador no tiene más que construir este dispositivo y conectarlo al port serie que ya está montado.

Descripción del dispositivo

El CI CRT 5027 contiene toda la circuitería lógica necesaria para originar todas las señales de tiempo (sincronismos vertical y horizontal, dirección de memoria para restauración de página, etc.) que precisa un terminal con TRC. Todos los detalles del formato íntegro de la representación, incluyendo entrelazado/no entrelazado, número de caracteres por fila, número de filas por cuadro, número de exploraciones por fila, anchura del impulso de sincronización horizontal y las señales de tiempo, son programables según las necesidades del usuario y para toda clase de formatos, tanto para todos los normalizados como para casi todos los no normalizados.

Si bien el CI CRT 5027 está estructurado básicamente para ser usado con su propio microprocesador, en este diseño se incluye un terminal ficticio formado por una PROM barata y su circuitería lógica tipo TTL para sustituir el control del microprocesador. Aunque así se aumenta al número de componentes, a cambio se consigue un terminal barato para caracteres y gráficos de gran calidad.

El CI CRT 8002 suministra una matriz de 7×11 puntos, una ROM generadora de 128 caracteres y un registro de desplazamiento en vídeo a alta velocidad para el cursor. Comprende la circuitería para originar las funciones tales como subrayado, parpadeo, vídeo inverso, borrado y penetración. La adición de modos de gráficos anchos y finos permite la formación de dibujos lineales, formas y símbolos gráficos originales.

* *Nota del traductor:* CRT, siglas en inglés de «cathode-ray tube» o lo que es igual, en español, «tubo de rayos catódicos», TRC.

Descripción del terminal

Como sucede con la mayoría de los proyectos electrónicos, un terminal con TRC implica un número grande de soluciones de compromiso entre eficacia y coste. Un formato de pantalla de 16 filas y 64 caracteres por fila minimiza los requisitos de memoria (1 kocteto) y mantiene la frecuencia de vídeo dentro del margen de los monitores de vídeo baratos, por lo que lo seleccionamos. Un formato de 80 caracteres por fila no sólo hubiera incrementado la frecuencia de vídeo más allá del ancho de banda de muchos monitores baratos, sino que, además, hubiera exigido más prestaciones de la memoria. De la misma forma, más filas por página hubieran requerido más capacidad de memoria, a no ser que se hubiese reducido el número de caracteres por fila.

En muchas aplicaciones de los microprocesadores, la memoria de página se comparte con el procesador por medio del bus de datos. En esta aplicación, la memoria de página es usada estrictamente por el TRC, con entrada de datos síncrona, carácter por carácter, a la posición del cursor.

Se pueden seleccionar gráficos y características del vídeo en la forma de carácter por carácter, utilizando palabras de control en el bus de datos de entrada. La figura 9.1 ilustra el diagrama de bloques del terminal.

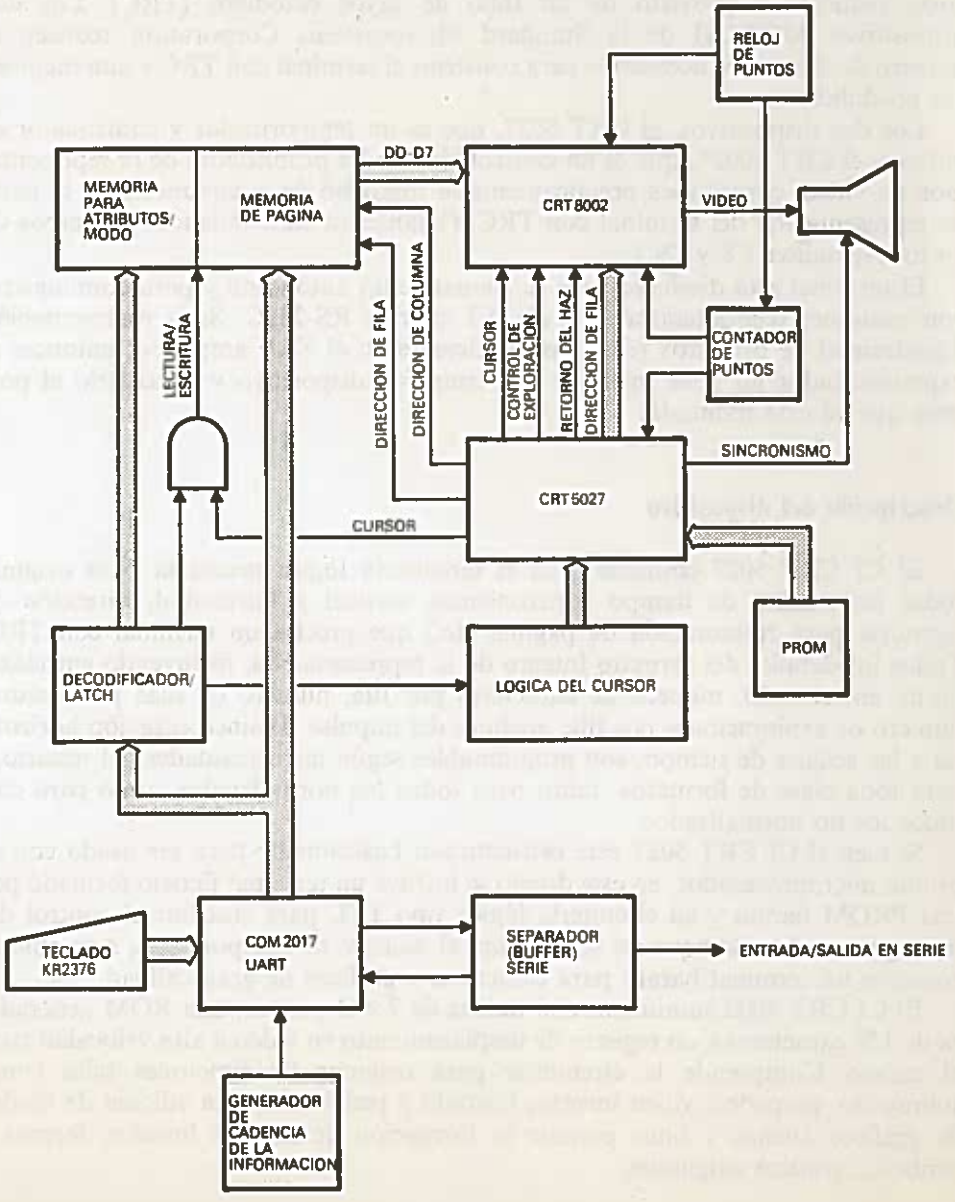


Figura 9.1 *Diagrama de bloques de un terminal barato con tubo de rayos catódicos.*

Formato de los caracteres

El CI CRT 8002 exige un bloque de caracteres 8×12, como mínimo, para formar su carácter básico de 7×11 y suministrar espaciado entre líneas y entre caracteres. Sin embargo, para lograr enmarcar completamente un carácter en una presentación de vídeo inverso, el bloque de caracteres horizontal ha de aumentarse a 9 ó 10 puntos. Por la misma razón, destinar 13 líneas a cada carácter permite enmarcarlo tanto por arriba como por abajo.

Con los barridos normalizados en TV de 60 Hz (vertical) y 15 750 Hz (horizontal), existen 15 750÷60=262,5 líneas por cuadro. Comoquiera que el funcionamiento sin entrelazado requiere un número par de líneas, se reduce la frecuencia horizontal a 15 720 Hz. De las 16 filas×13 líneas de exploración resultan 208 líneas de datos representados. Las restantes 54 líneas resultan borradas automáticamente por el CI CRT 5027 y suministran los márgenes superior e inferior.

Para tener en cuenta los márgenes derecho e izquierdo, así como el tiempo de retorno, se asigna un total de 80 caracteres a cada línea. Una buena regla para «andar por casa» es que el tiempo correspondiente al número total de caracteres sea un 25 % mayor que el correspondiente al número real de caracteres representados.

La frecuencia de vídeo se calcula como sigue: 10 (puntos por carácter)×80 (intervalos de un carácter por línea)×15 720 Hz (frecuencia de barrido horizontal)=12,576 MHz. Véanse las condiciones de trabajo en la tabla 9.1.

1. Matriz horizontal de caracteres (Número de puntos)	7
2. Matriz vertical de caracteres (Número de líneas de exploración vertical)	11
3. Bloque horizontal de caracteres (Paso 1 +espaciado horizontal deseado=Número de puntos)	10
4. Bloque vertical de caracteres (Paso 2+espaciado vertical deseado=Número de líneas de exploración horizontal)	13
5. Frecuencia (de restauración) de cuadros verticales (Frecuencia en Hz.)	60
6. Número deseado de filas de información	16
7. Número total de líneas de exploración activas en la «vídeo-representación» (Paso 4×Paso 6=Número de líneas de exploración horizontal)	208
8. Retardo del sincronismo vertical (Número de líneas de exploración horizontal)	26
9. Sincronismo vertical (Número de líneas de exploración horizontal; T=190.8 μs*)	3
10. Retardo de la exploración vertical (Número de líneas de exploración horizontal; T=1.59 ms*)	25
11. Cuadro vertical entero (Sumar los pasos del 7 al 10=Número de líneas de exploración horizontal)	262
12. Frecuencia de las líneas de exploración horizontal (Paso 5×Paso 11=Frecuencia en KHz)	15.720
13. Número deseado de caracteres por fila horizontal	64
14. Retardo del sincronismo horizontal (Número de intervalos de un carácter; T=4.77 μs**)	6
15. Sincronismo horizontal (Número de intervalos de un carácter; T=5.57 μs**)	7
16. Retardo de la exploración horizontal (Número de intervalos de un carácter; T=2.38 μs **)	3
17. Total de intervalos de un carácter en (1) línea de exploración horizontal (Sumar los pasos del 13 al 16)	80
18. Frecuencia de caracteres (Paso 12×Paso 17=Frecuencia en MHz)	1.2576
19. Frecuencia de puntos (Paso 3×Paso 18=Frecuencia en MHz)	12.576

* Intervalo vertical.
** Intervalo horizontal.

Tabla 9.1 *Características del CRT 5027 para un formato de pantalla de 64 caracteres por fila, 16 filas y sin entrelazado.*

REG. N.º	DIRECCION A3 A0	FUNCION	ASIGNACION DE BITS	HEX.	DEC.
0	0000	CONTAJE DE LINEAS HORIZONTALES 80	0 1 0 0 1 1 1 1	4F	79
1	0001	ENTRELAZADO 0 ANC. DEL SINC. HORIZ. 7 RET. DEL SINC. HORIZ. 6	0 0 1 1 1 1 1 0	3E	62
2	0010	EXPL. POR FILA DE INF. 13 CARACTERES POR FILA 64	X 1 1 0 0 0 1 1	63	99
3	0011	CARACTERES DE AJUSTE 1 FILAS DE INFORM. 16	1 0 0 0 1 1 1 1	8F	143
4	0100	EXPL. POR CUADRO 262 X = 3	0 0 0 0 0 0 1 1	03	3
5	0101	COMIENZO DE LA INFOR. VERTICAL =3+RETARDO DE LA EXPL. VERTICAL: RET. DE LA EXPL. 25 COMI. DE LA INF. 28	0 0 0 1 1 1 0 0	1C	28
6	0110	ULTIMA FILA REPRESENTADA (=FILAS DE INFORMACION)	X X 0 0 1 1 1 1	0F	15

Tabla 9.2 *Características de los registros de programación de un CRT 5027 para un formato de pantalla de 16×64.*

Programación del VTAC

El CI CRT 5027 (Video Timer And Controller, o sea, Temporizador y controlador de vídeo) es programable por el usuario en todos sus requisitos de tiempo y formatos. Los datos de programación están almacenados en 9 registros contenidos en un solo CI. Si bien el microprocesador puede suministrar fácilmente los datos de programación, en esta aplicación se emplea una PROM barata. Los 9 registros están programados como sigue (ver tabla 9.2).

Registro 0: Este registro contiene el número de intervalos de un carácter durante un período horizontal y suele ser 1,25 veces el número de caracteres por línea, en este caso $64 \times 1{,}25 = 80$. Como los contadores internos empiezan a contar desde cero, el número real contenido en el registro es $80 - 1 = 79$.

0	1	0	0	1	1	1	1

Registro 0

Registro 1: Tiene tres campos:

1) El bit 7 es «uno» si hay entrelazado y «cero» si no lo hay. En este ejemplo, se elige el funcionamiento no entrelazado.
2) Los bits del 3 al 6 programan el número de intervalos de un carácter durante la anchura del impulso de sincronismo horizontal. Este parámetro depende del monitor empleado y suele valer 5 µs. Comoquiera que hay 80 intervalos de un carácter en un período de exploración de 63,6 µs ($1 \div 15\,720$), cada intervalo dura 0,80 µs; siete intervalos de un carácter componen, pues, un impulso de 5,56 µs.
3) Los bits del 0 al 2 fijan el umbral anterior, es decir, sitúan la información horizontalmente. Las características del monitor determinarán la programación inicial, si bien puede ser necesario algún retoque experimental para centrar exactamente la información gráfica. Elegimos seis intervalos de un carácter para el umbral anterior.

0	0	1	1	1	1	1	0

Registro 1

Registro 2: Este registro posee dos campos:

1) Los bits del 3 al 6 (el bit 7 no se incluye) determinan el número de exploraciones por carácter. En nuestro caso, hemos definido los caracteres como de 10×13, por lo que se usa el equivalente binario de $13-1=12$ (todos los contadores del CI CRT 5027 comienzan a contar en «cero», no en «uno», por lo que su programación es siempre una unidad menor que el número).
2) Los bits del 0 al 2 contienen un código de 3 bits para el número de caracteres por línea. Según la hoja de características, el código para 64 es 011.

0	1	1	0	0	0	1	1

Registro 2

Registro 3: Este registro posee dos campos:

1) Los bits 6 y 7 retrasan el cursor de borrado y el tiempo de sincronismo para tener en cuenta los retardos debidos a la programación que experimentan el generador de caracteres y la memoria programable. Es normal que estos retrasos se compensen con el intervalo de un carácter.
2) Los bits del 0 al 5 definen el número de filas de datos, empezando desde el 0 binario, una vez más. Para una línea, se programará $16-1=15$.

1	0	0	0	1	1	1	1

Registro 3

Registro 4: El registro fija el número de tramas por cuadro. Para el modo sin entrelazado, esto se deduce de la fórmula $(N-256) \div 2=3$.

0	0	0	0	0	0	1	1

Registro 4

Registro 5: Este registro contiene el número de líneas por trama entre el arranque del impulso de sincronismo vertical y el comienzo de los datos (sincronismo vertical + margen posterior). Este tiempo ha de ser suficientemente largo para tener en cuenta todo el tiempo del retorno del monitor y permitir la situación vertical de la información gráfica. Usaremos aquí un 28. El umbral anterior es calculado por el CRT como $262-(13 \times 16)-28=26$.

0	0	0	1	1	1	0	0

Registro 5

Registro 6: El registro 6, o registro de enrollamiento («scrolling»), se programa con el número de la última fila de datos que se va a representar. Ya que deseamos establecer las condiciones básicas del CI CRT 5027, debemos programar este registro igual que el registro 3 (los bits 6 y 7 no se usan en este caso).

0	0	0	0	1	1	1	1

Registro 6

Registros 7 y 8: Estos dos registros contienen, respectivamente, el número del carácter y el número de la fila en que se encuentra el cursor. Dado que la posición inicial de éste es arriba a la izquierda, ambos registros contienen inicialmente ceros. Los cambios subsiguientes de la posición del cursor se introducirán en estos registros de la forma indicada en el apartado de «funcionamiento del circuito».

0	0	0	0	0	0	0	0

Registros 7 y 8

Descripción del circuito

En referencia a la figura 9.2, los CI 1A, 1B y 4 suministran el reloj de puntos de vídeo (12,58 MHz) y el de caracteres DCC, que es igual al de puntos dividido entre 10 (ya que cada carácter contiene 10 puntos a lo ancho). El reloj de puntos de vídeo determina la cadencia real de la información. El reloj de caracteres determina la rapidez con que los caracteres son solicitados. El CI 6A aísla la entrada del reloj de puntos del CI CRT 8002. Se utiliza una resistencia de enclavamiento a la salida para garantizar un «uno» lógico que se requiere a la entrada del VDC.

El mandato LOAD carga la información de los registros necesaria para programar el CRT 5027 a partir de la PROM CI 7 en el CRT 5027. La posibilidad de «autocarga» del CRT 5027 se utiliza para explorar, de forma automática, las direcciones de la PROM. La señal LOAD es producida automáticamente por el CI 1D al encender el circuito.

Dada la estructura del bus del CRT 5027, la información de la posición del cursor se carga en el mismo bus que los datos de los registros. Los selectores de datos triestables CI 14 y CI 15 seleccionan los datos de la posición X del cursor del contador CI 8 e CI 7, o los datos de la posición Y del cursor de CI 1D. Los CI 12 y CI 13 seleccionan el modo de direccionamiento para el CRT 5027. Se pueden usar tres modos: «autocarga sin procesador» para cargar desde los registros, carga de la posición X del cursor y carga de la posición Y del cursor.

Los circuitos CI 16 al CI 21 decodifican el modo de atributo y los controles del cursor del bus de datos ASCII. Si no se desean gráficos ni atributos especiales, los CI 16, 17 y 21 no son requeridos. Análogamente, si los controles del cursor están disponibles de forma directa, no es preciso decodificarlos.

Los CI 19 y CI 20 son PROMs de 256×4. Su programación exacta puede hacerse a la medida de las necesidades del usuario. La programación usada para este terminal se muestra en la tabla 9.3. Cuando se pulsa una tecla designada como atributo o clave de modo, la palabra de control adecuada queda retenida en el CI 21; todas las restantes entradas de información tendrán esta palabra impresa en los 4 bits superiores de la memoria programable. Ello permite que se pueda cambiar el modo o el atributo en la forma de carácter por carácter. El CI 18 es un decodificador de 2 a 4, que se activa cuando se decodifica un retroceso del cursor, un retroceso del carro o avance de línea, o una ↑ para suministrar el movimiento adeucado del cursor.

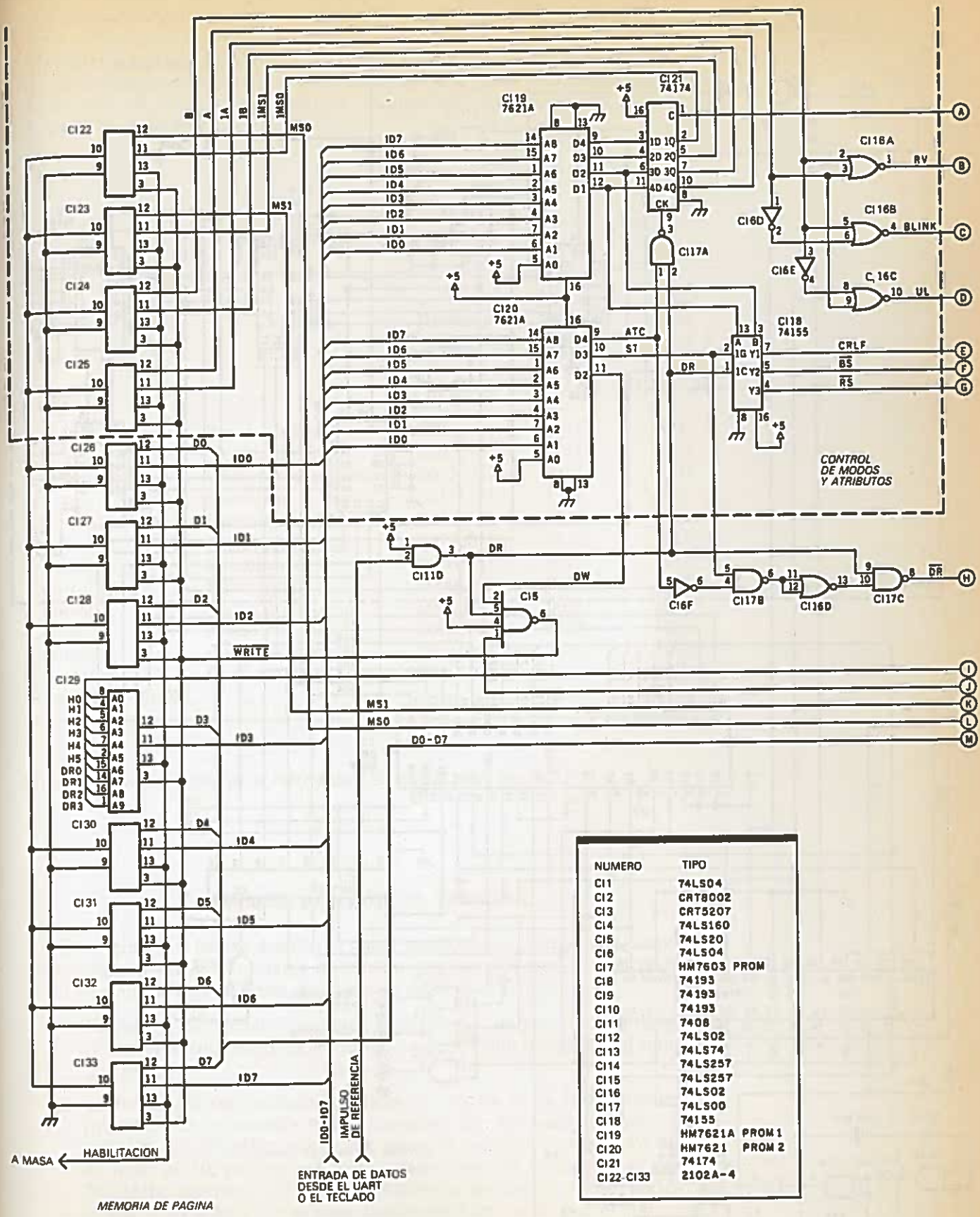
Se puede emplear circuitería TTL o TTL de bajo consumo en todo el esquema. Se recomienda emplear shottky TTL en el caso de CI 6 debido al breve tiempo de subida que exige la entrada del reloj.

Funcionamiento

Una vez encendido, hay que oprimir Control Q para retener el circuito en el modo «normal». Al pulsar las teclas de espacio y borrado simultáneamente, se borrará la pantalla. Cualquier entrada ulterior de caracteres será representada normalmente. Si se desean gráficos u otros atributos, se introducirá el código de control correspondiente. Este carácter no aparecerá en pantalla ni originará movimiento alguno del cursor, sino que se enganchará la nueva orden. Los modos se pueden cambiar para cada carácter solicitado. El movimiento del cursor puede decodificarse de la entrada en ASCII mediante la tecla de control, tal como indica la tabla 9.3.

El resto del circuito

La figura 9.3 ilustra la circuitería restante necesaria para realizar un terminal completo de entrada/salida en serie y compatible con el RS-232C. El empleo de elementos MOS/LSI reduce el número de pastillas a un mínimo indispensable.



NUMERO	TIPO
CI1	74LS04
CI2	CRT8002
CI3	CRT5207
CI4	74LS160
CI5	74LS20
CI6	74LS04
CI7	HM7603 PROM
CI8	74193
CI9	74193
CI10	74193
CI11	7408
CI12	74LS02
CI13	74LS74
CI14	74LS257
CI15	74LS257
CI16	74LS02
CI17	74LS00
CI18	74155
CI19	HM7621A PROM 1
CI20	HM7621 PROM 2
CI21	74174
CI22-CI33	2102A-4

Figura 9.2 *Esquema de un terminal barato y versátil con TRC utilizando las pastillas CRT 5027 y CRT 8002 (sigue en la página de enfrente).*

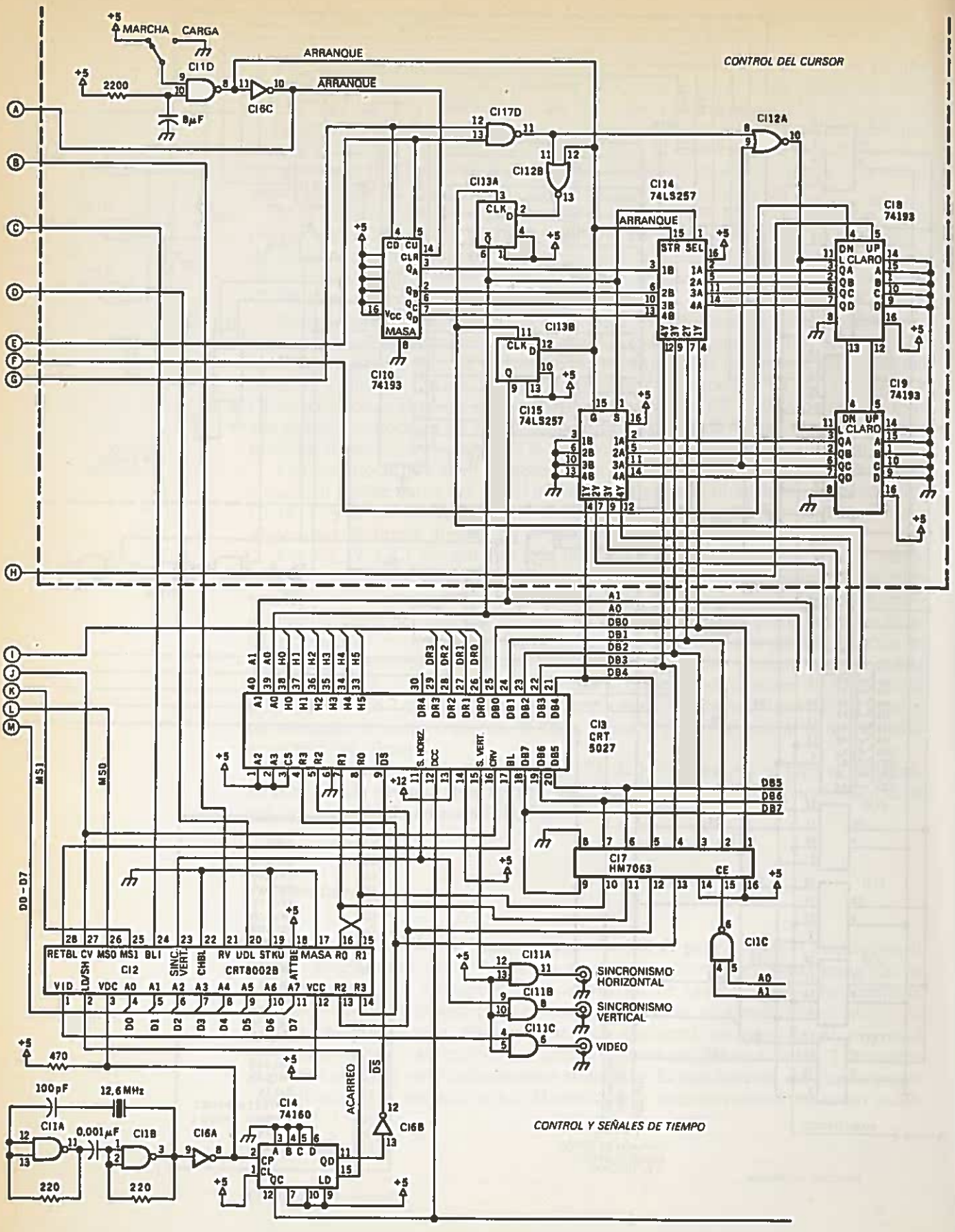


Figura 9.2 *(Continuación.)*

Un codificador de teclado KR2376, CI 1, codifica y elimina el rebote de las teclas y suministra una palabra de datos en ASCII al UART COM 2017 (véanse los Apéndices C6 y C7). El UART, a su vez, proporciona el enlace en serie para recepción/transmisión. El ritmo de la información se puede programar por medio de un código de entrada controlado por conmutador a un generador de cadencia de datos COM 8046 (ver Apéndice C10).

Programación de la PROM

Entrada de teclado	Función	Dirección	Salida de la PROM 1	Salida de la PROM 2
		76543210	$D_1D_2D_3D_4$	$D_1D_2D_3D_4$
Ret	Ret del carro	00011011	0011	1000
LF	Avance de línea	00010101	1011	1000
Control H	Cursor a la izquierda	00010001	0111	1000
RS	Cursor hacia arriba	00111101	1111	1000
US	Cursor a la derecha	00111111	1111	1010
Control Q	Atributo normal	00100011	1111	1011
Control W	Centelleo	00101111	1011	1011
Control E	Subrayado	00001011	0111	1011
Control R	Vídeo inverso	00100101	0011	1011
Control T	Modo externo	00101001	1101	1011
Control Y	Gráficos gruesos	00110011	1100	1011
Control U	Gráficos finos	00101011	1110	1011
Resto de la PROM			0011	1110

Tabla 9.3 *Programación de la PROM para el circuito de la figura 9.2.*

MODIFICACIONES AL TERMINAL

El terminal que hemos descrito se puede modificar fácilmente para adaptarlo a una amplia variedad de formatos de pantalla. Los cambios siguientes corresponden a un formato de 80 caracteres por fila y 24 filas:

1. Frecuencia de barrido horizontal; para permitir un aumento del número de líneas representadas (312), la frecuencia del barrido horizontal se aumenta a 20 220 Hz.
2. La frecuencia del oscilador de vídeo se calcula de la forma siguiente: 9 (puntos por carácter) × 100 (intervalos de un carácter por fila) × × 20 220 = 18,198 MHz. Adviértase que se han elegido 9 puntos por carácter en lugar de 10, ya que se hubiera ocasionado una frecuencia de vídeo de 20,2 MHz, superior a la frecuencia máxima a que trabaja el CRT 8002A. Así es que el CI 4 debe fijarse para dividir por 9 en vez de por 10.
3. Se requiere una memoria de página adicional de 1 K. La figura 9.4 muestra la modificación necesaria de las conexiones de direccionamiento.
4. La programación de los registros del CRT 5027 sigue las condiciones que muestran las tablas 9.4 y 9.5.

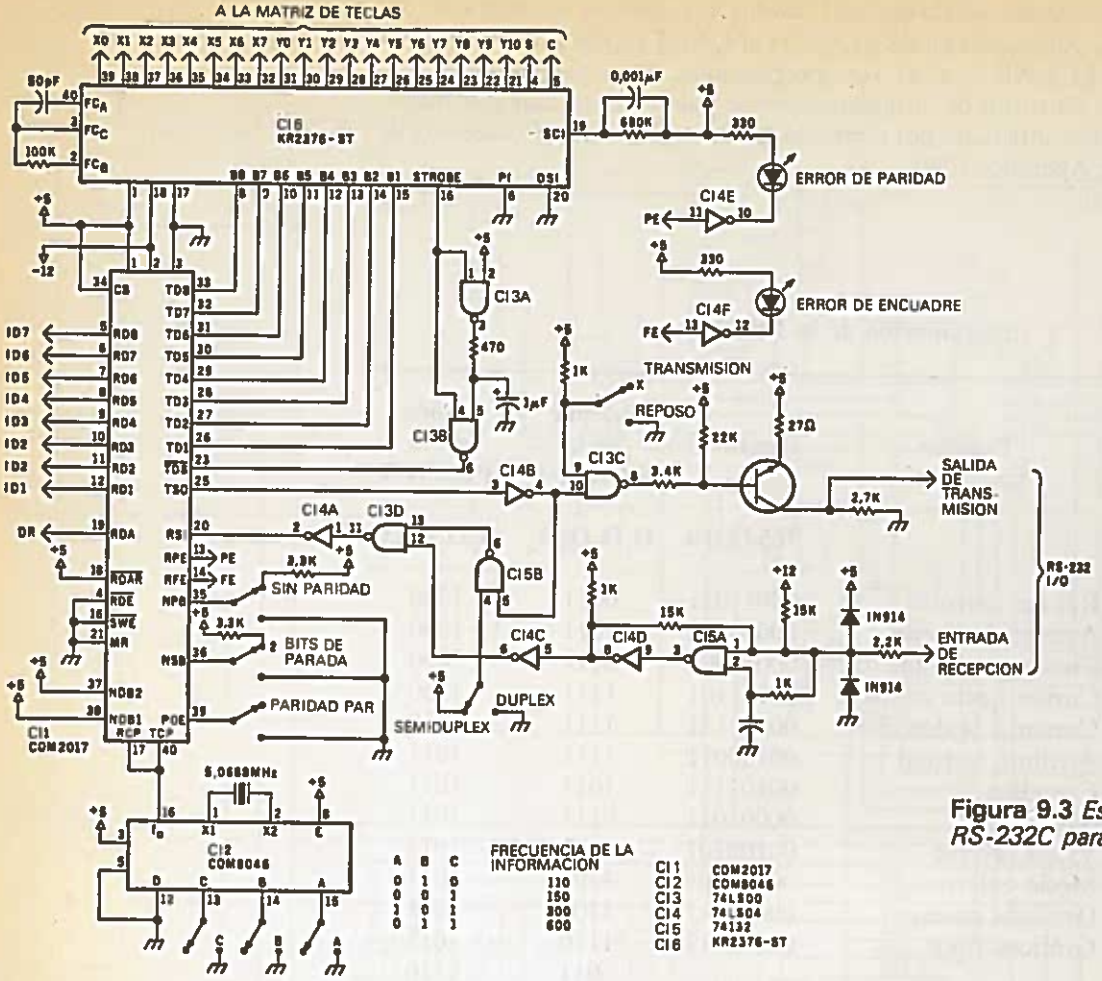


Figura 9.3 *Esquema del enlace tipo RS-232C para un terminal.*

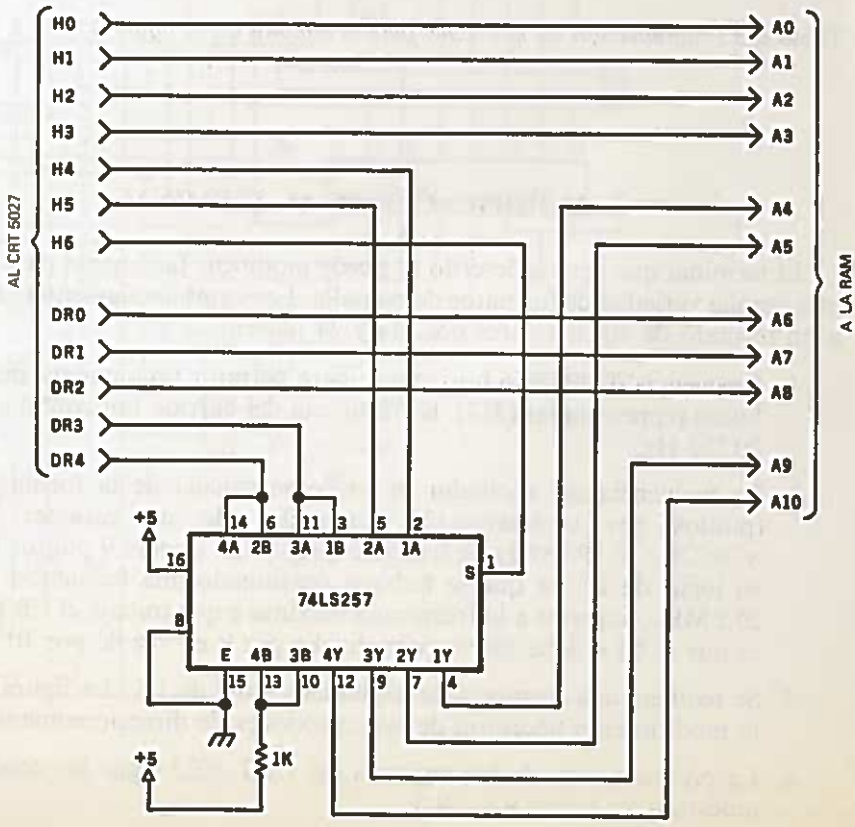


Figura 9.4 *Disposición de la memoria para un formato de pantalla de 24×80.*

1. Matriz horizontal de caracteres (Número de puntos)	7
2. Matriz vertical de caracteres (Número de líneas de exploración horizontal).	11
3. Bloque horizontal de caracteres (Paso 1 + Espaciado horizontal deseado = Número de puntos)	9
4. Bloque vertical de caracteres (Paso 2 + Espaciado vertical deseado = Número de líneas de exploración horizontal)	13
5. Frecuencia (de restauración) de cuadros verticales (Frecuencia en Hz	60
6. Número deseado de filas de información	24
7. Número total de líneas de exploración activas en la «vídeo-representación» (Paso 4 × Paso 6 = Número de líneas de exploración horizontal)	312
8. Retardo del sincronismo vertical (Número de líneas de exploración horizontal)	3
9. Sincronismo vertical (Número de líneas de exploración horizontal; T = 148.3 μs*)	3
10. Retardo de la exploración vertical (Número de líneas de exploración horizontal; R = 890.2 ms*)	18
11. Cuadro vertical entero (Sumar los pasos del 7 al 10 = Número de líneas de exploración horizontal)	336
12. Frecuencia de las líneas de exploración horizontal (Paso 5 × Paso 11. Frecuencia en kHz)	20220
13. Número deseado de caracteres por fila horizontal	80
14. Retardo del sincronismo horizontal (Número de intervalos de un carácter; T = 1.48 s**)	3
15. Sincronismo horizontal (Número de intervalos de un carácter); T = 4.94 μs**)	10
16. Retardo de la exploración horizontal (Número de intervalos de un carácter; T = 3.46 μs**)	7
17. Total de intervalos de un carácter en (1) línea de exploración horizontal (Sumar los pasos del 13 al 16)	100
18. Frecuencia de caracteres (Paso 12 × Paso 17 = Frecuencia en MHz)	2022
19. Frecuencia de puntos (Paso 3 × Paso 18 = Frecuencia en MHz)	18198

** Intervalo vertical.
** Intervalo horizontal.

Tabla 9.4 *Características del CRT 5027 para un formato de pantalla de 80 caracteres por fila, 24 filas y sin entrelazado.*

REG. N.º	DIRECCION A3 A0	FUNCION	ASIGNACION DE BITS	HEX.	DEC.
0	0000	CONTAJE DE LINEAS HORIZONTALES 100	0 1 1 0 0 0 1 1	63	99
1	0001	ENTRELAZADO 0 ANCHURA SINC. HORIZ. 10 RETARDO SINC. HORIZ. 3	0 1 0 1 0 0 1 1	53	83
2	0010	EXPL. FILA INFORM. 13 CARACT. POR FILA 80	X 1 1 0 0 1 0 1	65	101
3	0011	CARACT. DE AJUSTE 2 FILAS DE INFORM. 24	1 0 0 1 0 1 1 1	97	151
4	0100	EXPLOR. POR CUADRO 336 X = 40	0 0 1 0 1 0 0 0	28	40
5	0101	COMIENZO DE LA INF. VERT. = 3 + RETARDO EXPL. VERT.: RETARDO EXPL. 18 COMIENZO INF. 21	0 0 0 1 0 1 0 1	15	21
6	0110	ULTIMA FILA REPRESENT. (= FILAS DE INFORM.)	X X 0 1 0 1 1 1	17	23

Tabla 9.5 *Características de los registros de programación de un CRT 5027 para un formato de pantalla de 24×80.*

APENDICES

Apéndice A

Técnicas de montaje

RECOMENDACIONES PARA EL MONTAJE

Como resultado de haber montado un experimento todos los meses para mi columna «Ciarcia's Circuit Cellar» en la revista *BYTE* y por haber realizado también todos los circuitos de este libro, creo que puedo hablar como una autoridad sobre el tema del montaje de prototipos. Prototipo es un término de jerga que designa un montaje singular que se realiza a partir de un esquema, al contrario que un kit o experimento semimontado, que contiene un circuito impreso que sólo necesita instalarle los componentes.

No es fácil realizar prototipos. Existen muchos «así sí y así no», pero la realización satisfactoria depende fundamentalmente de la experiencia. Y la experiencia sólo se adquiere llevando cosas a la práctica.

Estas líneas se presentan con la intención de exponer este punto de vista. Sugiero al lector que comience con la fuente de alimentación, no sólo porque todo lo demás de la computadora no funciona sin ella, sino porque lleva incorporada la circuitería de protección, que será indulgente con el constructor si comete algún error. Además, al montar primeramente la fuente de alimentación, existe menos probabilidad de destruir el resto de la computadora al probar la propia fuente.

Por lo general, la regla de oro para realizar un prototipo es: ser pulcro. La computadora ZAP contiene frecuencias elevadas, por lo que su cableado debe seguir las distancias más cortas entre dos conexiones. Cuanto más largo sea el hilo, mejor funciona como antena. En casos extremos, la computadora puede llegar a dejar de funcionar debido al ruido inducido eléctricamente. Con las señales digitales relativamente lentas que portan los hilos ligados a los ports de entrada y salida, la situación se hace menos crítica. Los impulsos breves y los datos de alta velocidad, como las señales que se desarrollan en el control del procesador central y en las líneas de direccionamiento, son más críticas. En estos casos resulta siempre buena la idea de emplear circuitos de protección tales como las etapas separadoras.

Hasta un cierto grado, la computadora ZAP admite que sus componentes sean dispuestos de forma que encajen en el circuito impreso. La figura A.1. sugiere una manera: las conexiones pueden efectuarse enrollando o soldando los hilos a las patillas. Bastará cualquier placa capaz por su tamaño de alojar todas las pastillas, pero una buena elección podría ser la tarjeta normalizada S-100 especial para prototipos y disponible en la mayoría de los comercios. No existe otro bus que el de las señales estándar del Z-80, indicadas para ZAP, ya que se pretende ante todo que esté contenido en una sola tarjeta. El conector de 100 patillas dispone de capacidad suficiente para las conexiones de entrada/salida y c.c. Si el lector decide repartir el esquema entre más de una placa, hay que poner cuidado en efectuar la separación entre circuitos lógicos distintos; para mayor seguridad, se deben intercalar etapas separadas a la salida y a la entrada de las señales en las tarjetas, como, por ejemplo, es el caso de la memoria que puede ser montada en una placa aparte. Como se señaló en el texto, las líneas de direccionamiento y de datos necesarias para esta función están ya adecuadamente aisladas.

La cuestión de si es mejor arrollar o soldar los hilos en los terminales es una prerrogativa del montador. El autor prefiere personalmente las conexiones en firme de punto a punto, porque son más fáciles de modificar durante la localización de averías. La conexión por arrollamiento puede ser más fácil cuando ya se ha probado el circuito de ZAP y se lo ha rematado.

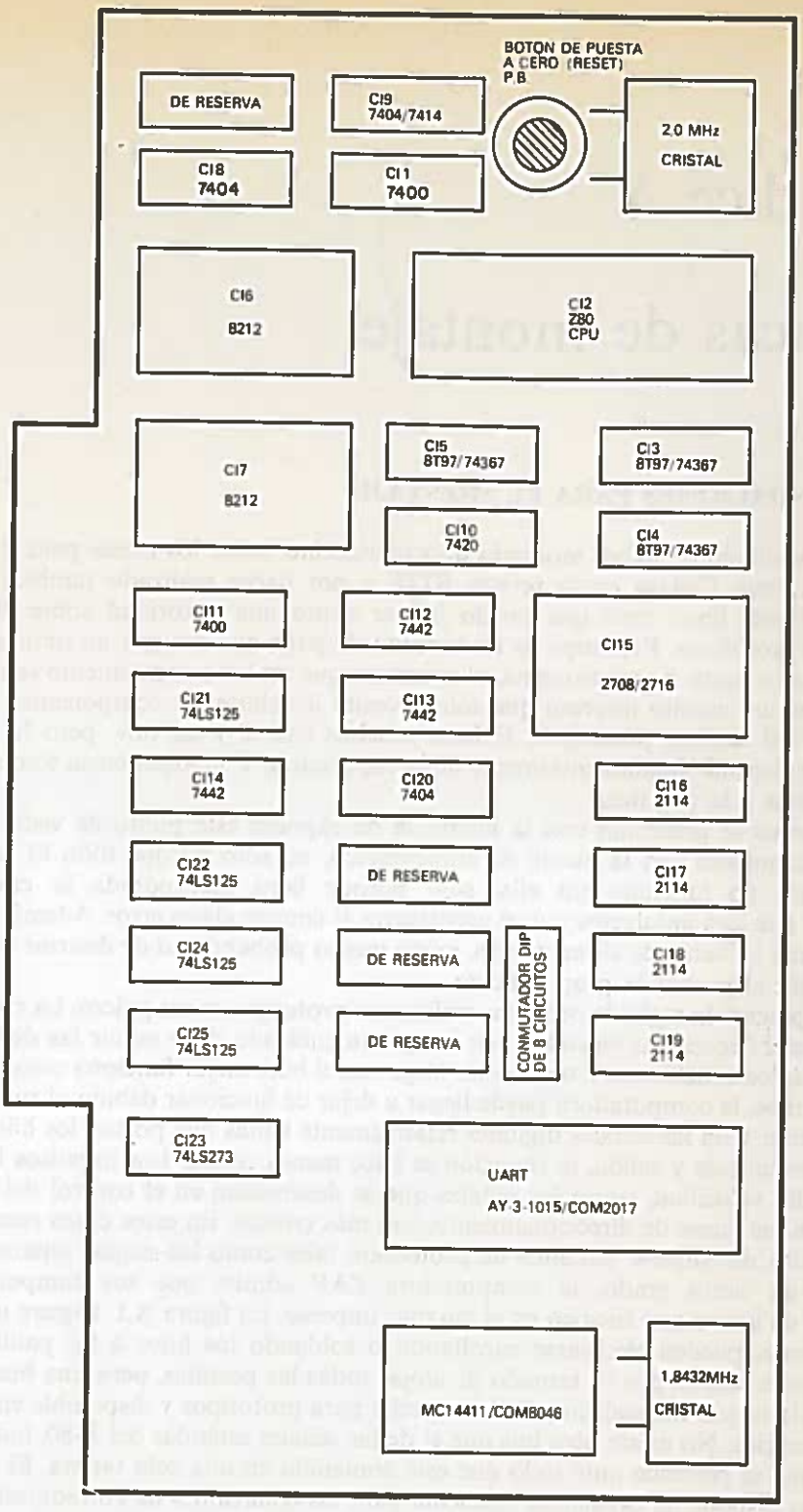


Figura A.1. *Disposición típica de la computadora ZAP básica.*

Se debe evitar que las conexiones de la fuente de alimentación sean demasiado largas y formen guirnaldas. Mejor que alambrar un hilo único que aporte los +5 V y otro hilo único conectado a masa, es usar una placa para prototipos con recubrimiento doble, de modo que el anverso y el reverso puedan estar conectados a masa y a los +5 V respectivamente. De esta forma, cada pastilla puede insertarse (usando zócalos para CI) y los cables de alimentación soldarse directamente a las superficies de cobre. Con conexiones

enrolladas o sin ellas, es buena idea soldar los cables de alimentación para reducir la posibilidad de conexiones intermitentes. Usar una superficie de cobre a masa para el cableado es uno de los mejores medios de reducir el ruido de las computadoras. Si la placa que posee el lector carece de una superficie de cobre que conectar a masa, entonces se puede soldar un cable grueso a lo largo del perímetro del circuito impreso y llevar a él hilos cortos.

Los condensadores de desacoplo son otro elemento indispensable para nuestro prototipo de computadora. Los circuitos integrados digitales, aun cuando son a prueba de calentamiento en la mayor parte de sus aplicaciones, son desgraciadamente sensibles al ruido que les llega mediante las conexiones de alimentación. A menudo esto les lleva a entrar en oscilación, por lo que es preciso colocar un condensador de 0,01 μ F a 0,1 μ F entre los +5 V y tierra cada tres CI para eliminar el problema. Otra idea buena consiste en situar un condensador electrolítico a la entrada de cada conexión de c.c de cada tarjeta. Por lo general, los condensadores serán de tántalo y se necesitarán 3 de ellos para otras tantas fuentes de alimentación de ZAP.

Finalmente, si le gusta la idea de la computadora ZAP, pero prefiere pasar el tiempo empleándola mejor que probándola, puede pensar en comparar las EPROMs programadas para el monitor de la ZAP. Este monitor para la computadora ZAP está disponible en forma de EMPROM 2708 o de EPROM 2716. Si se la prefiere de una sola tensión, por un precio módico, sírvase indicar el tipo que se desea cuando se haga el pedido a The Micromint, Inc., 917 Midway, Woodmere, Nueva York, 11598, Teléfono (516) 374-6793.

Apéndice B

Códigos ASCII

Decimal	Octal	Hexadecimal	Paridad, espacio	Carácter	Equiv. teclado control	Nombre de los códigos alternos
000	000	00	Par	NUL	@	CERO, CONTROL DESPLAZAMIENTO P, PRINCIPIO DE CINTA
001	001	01	Impar	SOH	A	PRINCIPIO DE CABECERA, SOM
002	002	02	Impar	STX	B	PRINCIPIO DE TEXTO, EOA
003	003	03	Par	ETX	C	FIN DE TEXTO, EOA
004	004	04	Impar	EOT	D	FIN DE LA TRANSMISION, END
005	005	05	Par	ENQ	E	PREGUNTA, WRU, ¿QUIEN ES USTED?
006	006	06	Par	ACK	F	ACUSE, RU, ¿QUIEN ES USTED?
007	007	07	Impar	BEL	G	CAMPANILLA
008	010	08	Impar	BS	H	RETROCESO, FEO
009	011	09	Par	HT	I	TABULADOR HORIZONTAL, TAB
010	012	0A	Par	LF	J	AVANCE DE LINEA, NUEVA LINEA, NL
011	013	0B	Impar	VT	K	TABULADOR VERTICAL, VTAB
012	014	0C	Par	FF	L	AVANCE DE PAGINA, FORMULARIO, PAGINA
013	015	0D	Impar	CR	M	RETORNO DEL CARRO, EOL
014	016	0E	Impar	SO	N	DESPLAZAMIENTO HACIA FUERA, RED SHIFT
015	017	0F	Par	SI	O	DESPLAZAMIENTO HACIA DENTRO, BLACK SHIFT
016	020	10	Impar	DLE	P	CAMBIO DE SIGNIFICADO DE DATOS, DC0
017	021	11	Par	DC1	Q	LECTOR EN MARCHA, XON
018	022	12	Par	DC2	R	PERFORADOR DE CINTA EN MARCHA
019	023	13	Impar	DC3	S	LECTOR PARADO, XOFF
020	024	14	Par	DC4	T	PERFORADOR DE CINTA PARADO
021	025	15	Impar	NAK	U	NO ACUSE, ERR
022	026	16	Impar	SYN	V	SINCRONISMO INACTIVO, SYNC
023	027	17	Par	ETB	W	FIN DE TEXTO, LEM
024	030	18	Par	CAN	X	CANCELACION, CANCL
025	031	19	Impar	EM	Y	FIN DEL MEDIO
026	032	1A	Impar	SUB	Z	SUSTITUIR
027	033	1B	Par	ESC	[	CAMBIO DE SIGNIFICADO, PREFIJO
028	034	1C	Impar	FS	\	SEPARADOR DE FICHEROS
029	035	1D	Par	GS	]	SEPARADOR DE GRUPOS
030	036	1E	Par	RS	^	SEPARADOR DE REGISTROS
031	037	1F	Impar	US	—	SEPARADOR DE UNIDADES
032	040	20	Impar	SP		ESPACIO, VACIO
033	041	21	Par	!		
034	042	22	Par	»		
035	043	23	Impar	#		
036	044	24	Par	$		
037	045	25	Impar	%		
038	046	26	Impar	&		

Decimal	Octal	Hexadecimal	Paridad, espacio	Carácter	Equiv. teclado control	Nombre de los códigos alternos
039	047	27	Par	'		APOSTROFO
040	050	28	Par	(		
041	051	29	Impar	)		
042	052	2A	Impar	*		
043	053	2B	Par	+		
044	054	2C	Impar	,		COMA
045	055	2D	Par	-		MENOS
046	056	2E	Par	.		
047	057	2F	Impar	/		
048	060	30	Par	0		NUMERO CERO
049	061	31	Impar	1		NUMERO UNO
050	062	32	Impar	2		
051	063	33	Par	3		
052	064	34	Impar	4		
053	065	35	Par	5		
054	066	36	Par	6		
055	067	37	Impar	7		
056	070	38	Impar	8		
057	071	39	Par	9		
058	072	3A	Par	:		
059	073	3B	Impar	;		
060	074	3C	Par	<		MENOR QUE
061	075	3D	Impar	=		
062	076	3E	Impar	>		MAYOR QUE
063	077	3F	Par	?		
064	100	40	Impar	@		DESPLAZAR P
065	101	41	Par	A		
066	102	42	Par	B		
067	103	43	Impar	C		
068	104	44	Par	D		
069	105	45	Impar	E		
070	106	46	Impar	F		
071	107	47	Par	G		
072	110	48	Par	H		
073	111	49	Impar	I		LETRA I
074	112	4A	Impar	J		
075	113	4B	Par	K		
076	114	4C	Impar	L		
077	115	4D	Par	M		
078	116	4E	Par	N		
079	117	4F	Impar	O		LETRA O
080	120	50	Par	P		
081	121	51	Impar	Q		
082	122	52	Impar	R		
083	123	53	Par	S		
084	124	54	Impar	T		
085	125	55	Par	U		
086	126	56	Par	V		
087	127	57	Impar	W		
088	130	58	Impar	X		
089	131	59	Par	Y		
090	132	5A	Par	Z		
091	133	5B	Impar	[		DESPLAZAR K
092	134	5C	Par	\		DESPLAZAR L
093	135	5D	Impar	]		DESPLAZAR M
094	136	5E	Impar	^		DESPLAZAR N
095	137	5F	Par	_		DESPLAZAR O, SUBRAYAR
096	140	60	Par	`		ACENTO GRAVE
097	141	61	Impar	a		
098	142	62	Impar	b		
099	143	63	Par	c		
100	144	64	Impar	d		

Decimal	Octal	Hexadecimal	Paridad, espacio	Carácter	Equiv. teclado control	Nombre de los códigos alternos
101	145	65	Par	e		
102	146	66	Par	f		
103	147	67	Impar	g		
104	150	68	Impar	h		
105	151	69	Par	i		
106	152	6A	Par	j		
107	153	6B	Impar	k		
108	154	6C	Par	l		
109	155	6D	Impar	m		
110	156	6E	Impar	n		
111	157	6F	Par	o		
112	160	70	Impar	p		
113	161	71	Par	q		
114	162	72	Par	r		
115	163	73	Impar	s		
116	164	74	Par	t		
117	165	75	Impar	u		
118	166	76	Impar	v		
119	167	77	Par	w		
120	170	78	Par	x		
121	171	79	Impar	y		
122	172	7A	Impar	z		
123	173	7B	Par	{		
124	174	7C	Impar	¦		CORTE VERTICAL
125	175	7D	Par	}		MODO ALTERNATIVO
126	176	7E	Par	~		(MODO ALTERNATIVO)
127	177	7F	Impar	DEL		BORRAR, RUBOUT

Apéndice C

Hojas de características técnicas según los fabricantes

Apéndice C1

intel

2708
MEMORIA PROM BORRABLE POR UV DE 8K (1K × 8)

	Potencia máxima	Acceso máximo
2708	800 mV	450 ns
2708L	425 mW	450 ns
2708-1	800 mW	350 ns
2708-6	800 mW	550 ns

- **Baja disipación de potencia - 425 mW. Máximo (2708L).**
- **Tiempo de acceso rápido - 350 ns. Máximo (2708-1).**
- **Estática - No necesita reloj.**
- **Entradas y salidas de datos compatibles con TTL durante ambos modos de lectura y programa.**
- **Salidas triestados - Posibilidad de interconexión OR.**

El Intel® 2708 es una EPROM de 8192 bits borrable mediante luz ultravioleta y eléctricamente reprogramable, idealmente adecuada donde haga falta inversión rápida y experimentación de patrones. Todas las entradas y salidas de datos son compatibles con TTL durante ambos modos de lectura y programa. Las salidas son triestados, permitiendo un interfaz directo con las estructuras de bus de los sistemas ordinarios.

La 2708L para 425 mW está disponible para aquellos sistemas que exijan menor disipación de energía que la de la 2708. Con la 2708L se obtiene un ahorro de más del 50 % en disipación de energía, sin menoscabo de la velocidad. La 2078L posee una alta inmunidad al ruido y está especificada con una tolerancia del 10 % para la fuente de alimentación. Existe también una 2708-1 de alta velocidad, 350 ns, para microprocesadores que requieran tiempos de acceso breves.

La familia 2708 se fabrica según la tecnología FAMOS con puertas de silicio sobre canal N y se encuentra disponible en cápsulas de 24 patillas tipo DIL.

CONFIGURACION DE LAS PATILLAS (PINES O TERMINALES)

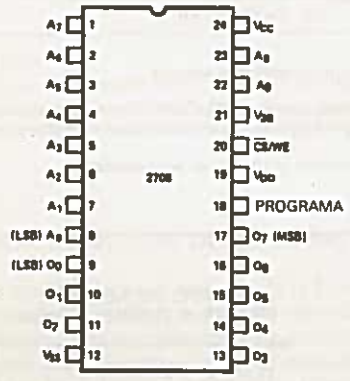
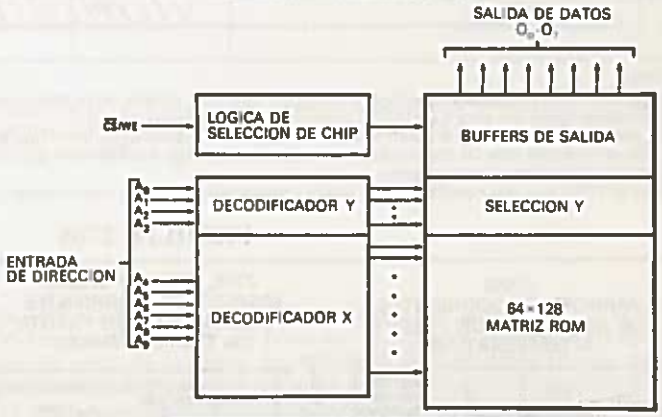


DIAGRAMA DE BLOQUES



NOMBRE DE LAS PATILLAS

A_0-A_9	ENTRADA DE DIRECCION
O_1-O_8	ENTRADAS/SALIDAS DE DATOS
$\overline{CS}$/WE	SELECCION DE CHIP/ENTRADA DE HABILITACION DE ESCRITURA

CONEXIONES DE LAS PATILLAS DURANTE LECTURA O PROGRAMA

MODO	NUMERO DE PATILLA							
	ENTRADA/SALIDA DE DATOS 9-11, 13-17	ENTRADA DE DIRECCION 1-8, 22-23	V_{SS} 12	PROGRAMA 18	V_{DD} 19	$\overline{CS}$/WE 20	V_{BB} 21	V_{CC} 24
LECTURA	D_{OUT}	A_{IN}	MASA	MASA	+12	V_{IL}	-5	+5
DESELECCION	ALTA IMPEDANCIA	IRRELEVANTES (NO IMPORTA)	MASA	MASA	+12	V_{IH}	-5	+5
PULSACION	D_{IN} a D_{OUT}	A_{IN}	MASA	PULSACION 26V	+12	V_{WW}	-5	+5

Impreso con permiso de Intel Corporation © 1980

PROGRAMACION

Las especificaciones de programación se describen en el apartado de Instrucciones de programación para PROM/ROM del Catálogo de Datos.

Regímenes máximos absolutos*

Temperatura bajo polarización	− 25° C a +85° C
Temperatura de almacenamiento	−65° C a +125° C
V_{DD} con respecto a V_{BB}	+20 V a −0,3 V
V_{CC} y V_{SS} con respecto a V_{BB}	+15 V a −0,3 V
Todas las tensiones de e/s con respecto a V_{BB} durante la lectura	+15 V a −0,3 V
Entrada $\overline{CS}/WE$ con respecto a V_{BB} durante la programación	+20 V a −0,3 V
Entrada de programa con respecto a V_{BB}	+35 V a −0,3 V
Disipación de potencia	1,5 W

*COMENTARIO: Los excesos por encima de los valores indicados bajo los «regímenes máximos absolutos» pueden ocasionar daños permanentes al dispositivo. Esto es un régimen de exceso solamente y el funcionamiento del dispositivo no está implicado bajo estas u otras condiciones por encima de las indicadas en el apartado operativo de estas especificaciones. La exposición durante largos períodos a las condiciones de los regímenes máximos absolutos puede afectar a la fiabilidad del dispositivo.

CONDICIONES DE FUNCIONAMIENTO EN C.C. Y C.A. DURANTE LA LECTURA

	2708	2708-1	2708-6	2708L
Margen de temperatura	0°C−70°C	0°C−70°C	0°C−70°C	0°C−70°C
Fuente de alimentación V_{CC}	5 V±5%	5 V±5%	5 V±5%	5 V±10%
Fuente de alimentación V_{DD}	12 V±5%	12 V±5%	12 V±5%	12 V±10%
Fuente de alimentación V_{BB}	−5 V±5%	−5 V±5%	−5 V±5%	−5 V±10%

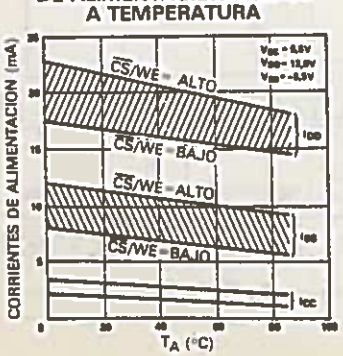
FUNCION DE LECTURA CARACTERISTICAS DE FUNCIONAMIENTO EN C.C. Y C.A.

Símbolo	Parámetro	2708, 2708-1, 2708-6 Límites			2708L Límites			Unidades	Condiciones de prueba
		Mín.	Típ.[2]	Máx.	Mín.	Típ.[2]	Máx.		
I_{LI}	Corriente de retorno a la entrada de dirección y selección de chip		1	10		1	10	μA	V_{IN}=5,25 V o $V_{IN}=V_{IL}$
I_{LO}	Corriente de fuga a la salida		1	10		1	10	μA	V_{OUT}=5,5 V, CS/WE=5 V
I_{DD}[3]	Corriente de alimentación de V_{DD}		50	65		21	28	mA	Corrientes de alimentación en el caso más desfavorable[4]
I_{CC}[3]	Corriente de alimentación de V_{CC}		6	10		2	4	mA	Todas las entradas a nivel alto
I_{BB}[3]	Corriente de alimentación de V_{BB}		30	45		10	14	mA	CSWE=5 V; T_A=0 C
V_{IL}	Tensión de nivel bajo a la entrada	V_{SS}		0,65	V_{SS}		0.65	V	
V_{IH}	Tensión de nivel alto a la entrada	3,0		V_{CC}+1	2,2		V_{CC}+1	V	
V_{OL}	Tensión de nivel bajo a la salida			0,45			0,4	V	I_{OL}=1,6 mA (2708, 2708-1, 2708-6)
									I_{OL}=2 mA (2708L)
V_{OH1}	Tensión de nivel alto a la salida	3,7			3,7			V	I_{OH}=−100μA
V_{OH2}	Tensión de nivel alto a la salida	2,4			2,4			V	I_{OH}=−1mA
PD	Disipación de potencia			800			325	mW	T_A=70 C
							425	mW	T_A=0 C

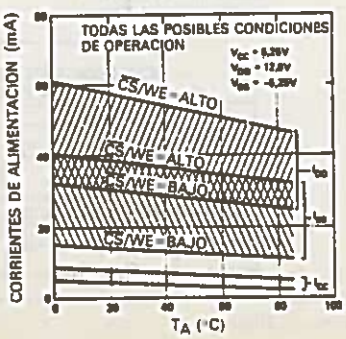
NOTAS:
1. V_{BB} debe estar aplicada antes que V_{CC} y que V_{DD}. V_{BB} debe, además, ser la última fuente de alimentación apagada.
2. Los valores típicos con para T_A=15 °C y tensiones de alimentación nominales.
3. La potencia total disipada no se calcula sumando las diferentes corrientes (I_{DD}, I_{CC} e I_{BB}) multiplicadas por sus tensiones respectivas, ya que existen corrientes comunes entre las distintas fuentes de alimentación y V_{SS}. Las corrientes I_{DD}, I_{CC} e I_{BB} deben usarse únicamente para determinar la capacidad de la fuente de alimentación.
4. Para la 2708L I_{BB} está especificada en el estado «programado» y alcanza como máximo 18 mA en el estado «no programada».

FAMILIA 2708

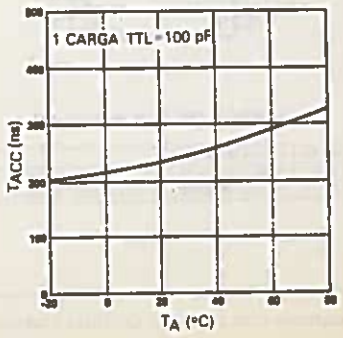
2708L MARGEN DE CORRIENTES DE ALIMENTACION FRENTE A TEMPERATURA



2708, 2708-1 Y 2708-6 MARGEN DE CORRIENTES DE ALIMENTACION FRENTE A TEMPERATURA



TIEMPO DE ACCESO FRENTE A TEMPERATURA



CARACTERISTICAS EN C.A.

Símbolo	Parámetro	2708, 2708L Límites		2708-1 Límites		2708-6 Límites		Unidades
		Mín.	Máx.	Mín.	Máx.	Mín.	Máx.	
t_{ACC}	Retardo entre dirección y salida		450		350		550	ns
t_{CO}	Retardo entre selección de chip y salida		120		120		160	ns
t_{DF}	No selección de chip a salida flotante	0	120	0	120	0	160	ns
t_{OH}	Dirección a retención de salida	0		0		0		ns

CAPITANCIA[1] $T_A = 25°C$, f = 1 MHz

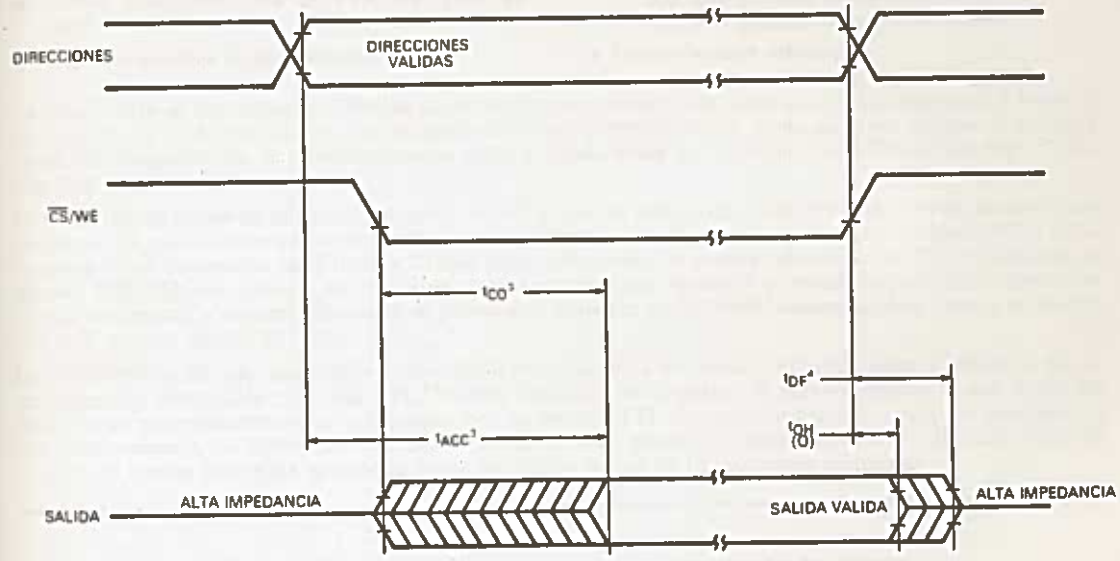
Símbolo	Parámetro	Típ.	Máx.	Unidad	Condiciones
C_{IN}	Capacitancia de entrada	4	6	pF	$V_{IN} = 0V$
C_{OUT}	Capacitancia de salida	8	12	pF	$V_{OUT} = 0V$

NOTA:
[1] Este parámetro es muestreado periódicamente y no está probado al 100 %.

CONDICIONES DE PRUEBA EN C.A.

Carga de salida: 1 puerta TTL y $C_L = 100$ pF.
Tiempos de subida y bajada a la entrada: ≤ 20 ns.
Niveles de referencia para las medidas de tiempo: 0,8 y 2,8 V para la entrada; 0,8 y 2,4 V para las salidas.
Niveles de los impulsos de entrada % de 0,65 a 3,0 V.

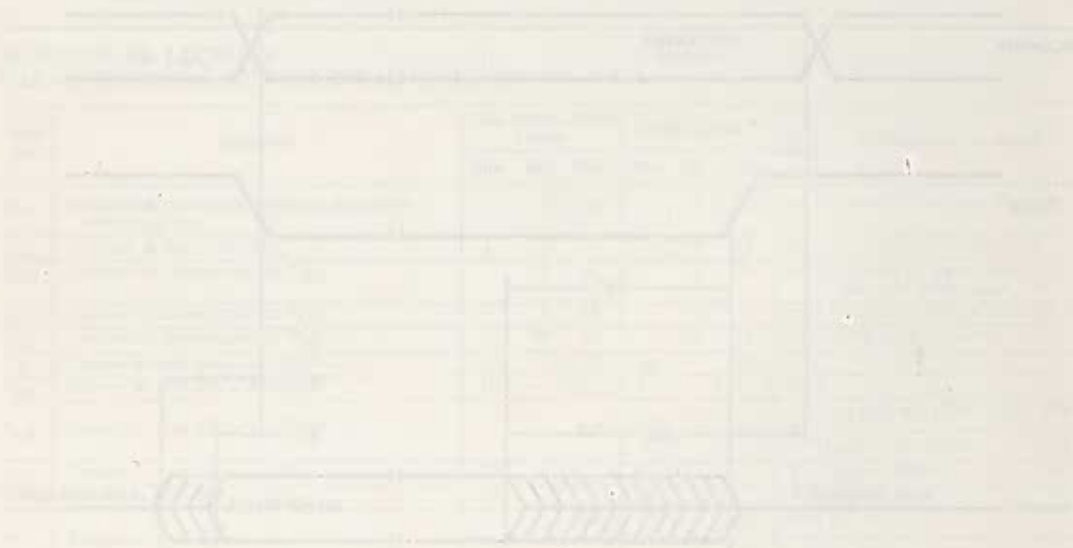
FORMAS DE ONDA EN C.A.[2]



NOTAS:
2. Todos los tiempos entre paréntesis son mínimos y están en nanosegundos, mientras no se indique lo contrario.
3. $\overline{CS}$ puede estar retrasada hasta ($t_{ACC} - t_{CO}$) después que las direcciones son válidas, sin afectar a t_{ACC}.
4. t_{DF} depende de CS o del cambio de dirección, lo que ocurra antes.

CARACTERISTICAS DE BORRADO

Las características de borrado de la familia 2708 son tales que esta acción comienza con la exposición a la luz de longitudes de onda menores que unos 4000 Angstroms (Å). Hay que notar que la luz solar y la de ciertos tipos de lámparas fluorescentes contiene longitudes de onda de la gama de 3000-4000 Å. Los datos revelan que la exposición constante al nivel de iluminación fluorescente de una habitación podría borrar una memoria de este tipo en aproximadamente 3 años, mientras que la exposición directa a la luz solar lo haría en una semana. Intel dispone de etiquetas opacas para los casos en que la 2708 haya de ser expuesta a estas condiciones de iluminación durante periodos de tiempo prolongados. Estas etiquetas se deben colocar sobre las ventanas de la 2708 para evitar su borrado involuntario.
El procedimiento recomendado para el borrado (véase el Apartado de Instrucciones de Programación de PROM/ROM del Catálogo de Datos) de la familia 2708 consiste en exponerla a la luz ultravioleta que contenga la longitud de onda de 2537 Å. La dosis integral (esto es, intensidad UV × tiempo de exposición) para el borrado debe ser como mínimo de 15 W-seg/cm^2. El tiempo de borrado para esta dosis es aproximadamente de 15 a 20 minutos, utilizando una lámpara ultravioleta de régimen de potencia de 1200 μW/cm^2. El dispositivo debe colocarse a menos de 2,5 cm de los tubos de la lámpara durante el borrado. Algunas lámparas poseen un filtro en el tubo, que es preciso retirar antes del borrado.



Apéndice C2

intel

2716

MEMORIA PROM BORRABLE POR UV DE 16K (2K × 8)

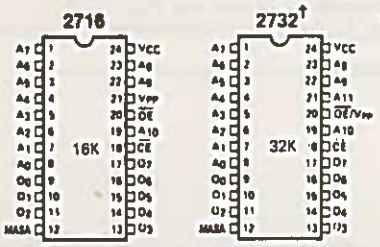
- **Tiempo de acceso rápido:**
  - **350 ns máx. 2716-1.**
  - **390 ns máx. 2716-2.**
  - **450 ns máx. 2716.**
  - **490 ns máx. 2716-5.**
  - **650 ns máx. 2716-6.**
- **Patillas compatibles con el EPROM 2732 de Intel®.**
- **Requisitos sencillos de programación:**
  - **Programación con posicionamiento único.**
  - **Programas con un impulso de 50 ms.**
- **Alimentación de +5 V únicamente.**
- **Entradas y salidas compatibles con TTL durante lectura y programa.**
- **Baja disipación de potencia:**
  - **525 mW potencia máxima en activo.**
  - **132 mW potencia máxima en reserva.**
- **Completamente estática.**

La Intel® 2716 es una memoria EPROM de 16 384 bits para lectura sólo, programable eléctricamente y borrable con luz UV. La 2716 funciona con una fuente de alimentación sencilla de 5 V, posee un modo de reserva estático y ofrece una programación de direccionamiento único y rápido. Hace que el diseño con EPROM sea más rápido, más fácil y más económico.

La 2716, con su fuente de alimentación única de 5 V y con un tiempo de acceso de hasta 350 ns, es ideal para usarla con los nuevos microprocesadores de elevada eficiencia, alimentados a 5 V, tal como los Intel 8085 y 8086. También están disponibles las 2716-5 y 2716-6 para aplicaciones de menos velocidad. La 2716 es, además, la primera EPROM que presenta un modo de reserva estático que reduce la potencia disipada sin aumentar el tiempo de acceso. La máxima disipación de potencia en activo es de 525 mW, mientras que en reserva es de sólo 132 mW, con un ahorro del 75 %.

La 2716 ofrece el método más simple y más rápido conocido en la actualidad para programar EPROM, o sea, la programación monopulso con nivel TTL. No hay necesidad de impulsos de mayor tensión, ya que todos los controles de programación están gobernados por las señales TTL. Se puede programar cualquier dirección en cualquier momento, sea individual, secuencial o aleatoriamente, gracias a la programación con dirección única de la 2716. El tiempo total para programar todos los 16 384 bits es de 100 segundos solamente.

CONFIGURACION DE PATILLAS



† REFIERASE A LAS HOJAS DE DATOS DE LA 2732 PARA VER LAS CARACTERISTICAS

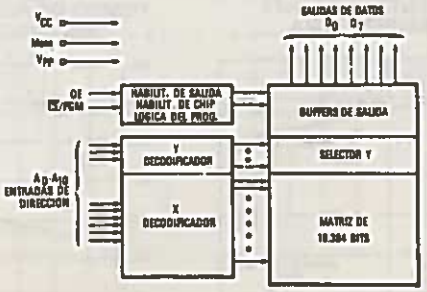
SELECCION DE MODOS

MODO \ PATILLA	$\overline{CE}$/PGM (18)	$\overline{OE}$ (20)	V_{PP} (21)	V_{CC} (24)	SALIDAS (9-11, 13-17)
Lectura	V_{IL}	V_{IL}	+5	+5	D_{OUT}
Reserva	V_{IH}	Indiferente	+5	+5	Z alto
Programa	Pulso desde V_{IL} hasta V_{IH}	V_{IH}	+25	+5	D_{IN}
Verificación del programa	V_{IL}	V_{IL}	+25	+5	D_{OUT}
Inhibición del programa	V_{IL}	V_{IH}	+25	+5	Z alto

NOMBRE DE LAS PATILLAS

A_0–A_{10}	DIRECCIONES
$\overline{CE}$/PGM	HABILITACION DE CHIP/PROGRAMA
$\overline{OE}$	HABILITACION DE SALIDAS
O_0–O_7	SALIDAS

DIAGRAMAS DE BLOQUES



Impreso con permiso de Intel Corporation © 1980.

PROGRAMACION

Las características de programación se describen en el Apartado de Instrucciones para la Programación de PROM/ROM del Catálogo de Datos.

Regímenes máximos absolutos*

Temperatura bajo polarización ...	−10°C a +80°C
Temperatura de almacenamiento..	−65°C a +125°C
Todas las tensiones de entrada o salida con respecto a tierra	+6 V a 0,3 V
Tensión de alimentación V_{PP} con respecto a tierra	+26,5 V a −0,3 V

*COMENTARIO: Los excesos por encima de los valores indicados bajo los regímenes máximos absolutos pueden ocasionar daños permanentes al dispositivo. Esto es un régimen de exceso solamente y el funcionamiento del dispositivo no está implicado bajo estas u otras condiciones por encima de las indicadas en el apartado operativo de estas especificaciones. La exposición durante largos períodos a las condiciones de los regímenes máximos absolutos puede afectar a la fiabilidad del dispositivo.

Condiciones de funcionamiento en c.c. y c.a. durante la lectura

	2716	2716-1	2716-2	2716-5	2716-6
Margen de temperatura	0°C−70°C	0°C−70°C	0°C−70°C	0°C−70°C	0°C−70°C
Tensión continua de alimentación V_{CC} (1,2)	5 V±5%	5 V±10%	5 V±5%	5 V±5%	5 V±5%
Tensión continua de alimentación V_{PP} (2)	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}

FUNCIONAMIENTO EN LECTURA

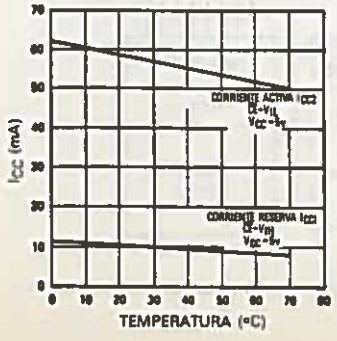
Corriente continua y características de funcionamiento

Símbolo	Parámetro	Límites			Unidad	Condiciones
		Mín.	Tip.[3]	Máx.		
I_{LI}	Corriente de carga a la entrada			10	μA	$V_{IN}=5{,}25$ V
I_{LO}	Corriente de fuga a la salida			10	μA	$V_{OUT}=5{,}25$ V
I_{PP1}[2]	Corriente V_{PP}			5	mA	$V_{PP}=5{,}25$ V
I_{CC1}[2]	Corriente V_{CC} (reserva)		10	25	mA	$\overline{CE}=V_{IH}$, $\overline{OE}=V_{IL}$
I_{CC2}[2]	Corriente V_{CC} (activada)		57	100	mA	$\overline{OE}=\overline{CE}=V_{IL}$
V_{IL}	Tensión de entrada para nivel bajo	−0,1		0,8	V	
V_{IH}	Tensión de entrada para nivel alto	2,0		$V_{CC}+1$	V	
V_{OL}	Tensión de salida para nivel bajo			0,45	V	$I_{OL}=2{,}1$ mA
V_{OH}	Tensión de salida para nivel alto	2,4			V	$I_{OH}=-400$ μA

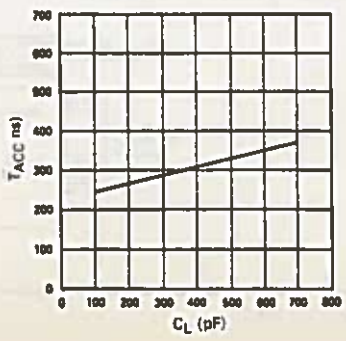
NOTAS:
1. V_{CC} debe ser aplicada simultáneamente o antes que V_{PP} y retirada simultáneamente o después que V_{PP}.
2. V_{PP} puede conectarse directamente a V_{CC} excepto durante la programación. La corriente de alimentación sería entonces la suma de I_{CC} e I_{PP1}.
3. Los valores típicos son para $T_A=25$°C y tensiones de alimentación nominales.
4. Este parámetro se muestrea nada más y no se prueba al 100%.

Características típicas

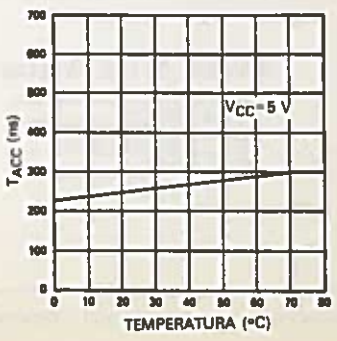
CORRIENTE I_{CC} FRENTE A TEMPERATURA



TIEMPO DE ACCESO FRENTE A CAPACITANCIA



TIEMPO DE ACCESO FRENTE A TEMPERATURA



CARACTERISTICAS EN C.A.

Símbolo	Parámetro	Límites (ns)										Condiciones de prueba
		2716		2716-1		2716-2		2716-5		2716-6		
		Mín.	Máx.	Mín.	Máx.	Mín.	Máx.	Mín.	Máx.	Mín.	Máx.	
t_{ACC}	Retraso entre dirección y salida		450		350		390		450		450	$\overline{CE}=\overline{OE}=V_{IL}$
t_{CE}	Retraso entre $\overline{CE}$ y salida		450		350		390		490		650	$\overline{OE}=V_{IL}$
t_{OE}	Retraso entre habilitación de salida y salida		120		120		120		160		200	$\overline{CE}=V_{IL}$
t_{DF}	De nivel alto para la habilitación de salida a salida flotante	0	100	0	100	0	100	0	100	0	100	$\overline{CE}=V_{IL}$
t_{OH}	Retención de la salida desde las direcciones, $\overline{CE}$ u $\overline{OE}$, cualquiera que ocurra la primera	0		0		0		0		0		$\overline{CE}=\overline{OE}=V_{IL}$

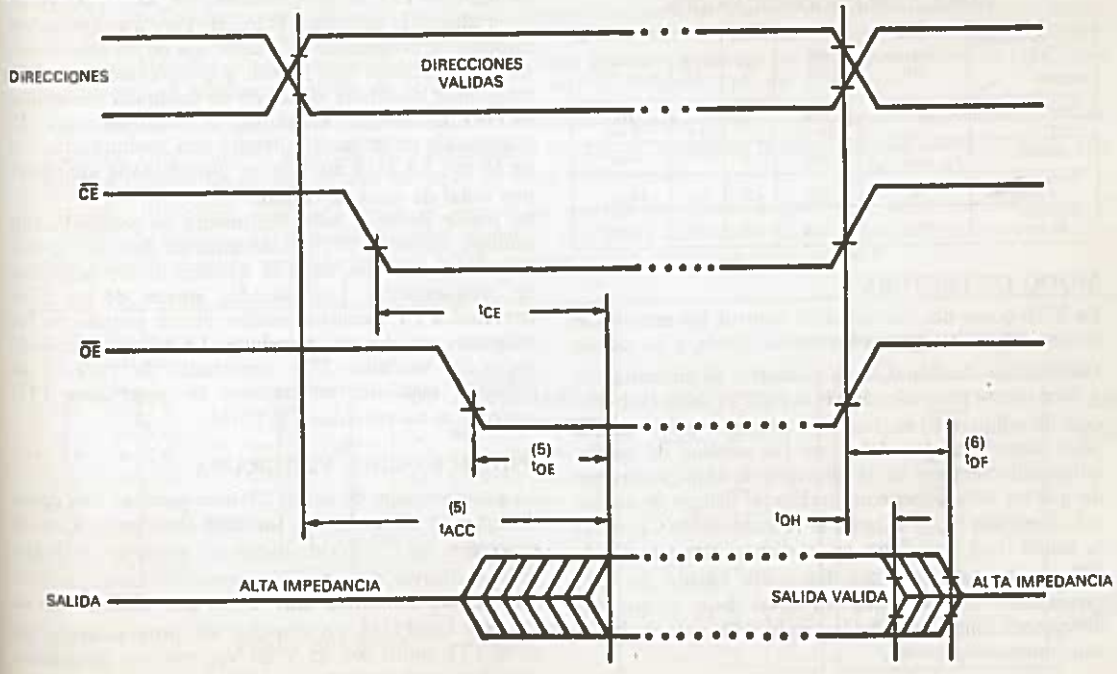
Capacitancia[4] $T_A=25°C$, f=1 MHz

Símbolo	Parámetro	Típ.	Máx.	Unidad	Condiciones
C_{IN}	Capacitancia de entrada	4	6	pF	$V_{IN}=0V$
C_{OUT}	Capacitancia de salida	8	12	pF	$V_{OUT}=0V$

Condiciones de prueba en c.a.

Carga de salida: 1 puerta TTL y una capacidad $C_L=100$ pF.
Tiempos de subida y bajada para la entrada ≤ 20 ns.
Niveles del impulso de entrada: 0,8 a 2,2 V.
Nivel de referencia para la medida del tiempo.
Entradas 1 y 2 V.
Salidas 0,8 y 2 V.

Formas de onda[1]



NOTAS
1. V_{CC} debe ser aplicada simultáneamente o antes que V_{PP} y retirada simultáneamente o después que V_{PP}.
2. V_{PP} puede conectarse directamente a V_{CC} excepto durante la programación. La corriente de alimentación sería entonces la suma de I_{CC} e I_{PP1}.
3. Los valores típicos son para $T_A=25°C$ y tensiones de alimentación nominales.
4. Este parámetro se muestrea nada más y no se prueba al 100 %.
5. OE puede estar retrasada hasta $t_{ACC}-t_{OE}$ tras el flanco descendente de CE sin afectar a t_{ACC}.
6. t_{DF} está determinada por OE o CE, la que ocurra antes.

CARACTERISTICAS DE BORRADO

Las características de borrado de la 2716 son tales que el borrado comienza en cuanto se la expone a luz de longitudes de onda más cortas que unos 4000 Angstroms (Å). Hay que notar que la luz solar y la de algunos tipos de lámparas fluorescentes contiene longitudes de onda dentro de la gama de 3000 a 4000 Å. Los datos revelan que la exposición constante al nivel de iluminación fluorescente de una habitación podría borrar una 2716 típica en aproximadamente 3 años, mientras que la exposición directa a la luz solar lo haría en una semana. Intel dispone de etiquetas opacas para los casos en que la 2716 haya de ser expuesta a estas condiciones de iluminación durante períodos prolongados de tiempo. Estas etiquetas se deben colocar sobre la ventana de la 2716 para evitar su borrado involuntario.

El procedimiento recomendado para el borrado (véase el Apartado de Instrucciones de Programación de PROM/ROM del Catálogo de Datos) de la 2716 es exponerla a la luz ultravioleta de corta longitud de onda, especialmente la de 2537 Å. La dosis integrada (esto es, intensidad UV × tiempo de exposición) para el borrado debe ser como mínimo de 15 W-seg/cm^2. El tiempo de borrado para esta dosis es aproximadamente de 15 a 20 minutos, utilizando una lámpara ultravioleta de régimen de potencia de 1200 μW/cm^2. La 2716 debe colocarse a menos de 2,5 cm de los tubos de la lámpara durante el borrado. Algunas lámparas poseen filtro en el tubo, que es preciso retirar antes del borrado.

FUNCIONAMIENTO DEL DISPOSITIVO

Los cinco modos de funcionamiento de la 2716 están enumerados en la tabla I. Nótese que todas las entradas para los cinco modos son niveles TTL. Las tensiones de alimentación son $V_{CC} = +5$ V y una V_{PP}. La fuente V_{PP} debe suministrar 25 V durante tres modos de programación y 5 V en los otros dos.

TABLA I. SELECCION DE MODOS

PATILLAS / MODO	$\overline{CE}$/PGM (18)	$\overline{OE}$ (20)	V_{PP} (21)	V_{CC} (24)	SALIDAS (9-11, 13-17)
Lectura	V_{IL}	V_{IL}	+5	+5	D_{OUT}
Reserva	V_{IH}	Irrelevante	+5	+5	Z alta
Programa	Pulsos desde V_{IL} hasta V_{IH}	V_{IH}	+25	+5	D_{IN}
Verificación del programa	V_{IL}	V_{IL}	+25	+5	D_{OUT}
Inhibición del programa	V_{IL}	V_{IH}	+25	+5	Z alta

MODO DE LECTURA

La 2716 posee dos funciones de control, las cuales han de ser satisfechas para obtener los datos a su salida.

Habilitación de chip (CE) es el control de alimentación y debe usarse para seleccionar la patilla (chip). Habilitación de salida ($\overline{OE}$) es el control de salida y debe usarse para seleccionar los datos en las patillas de salida, independientemente de la selección de chip. Suponiendo que las direcciones son estables, el tiempo de acceso a la dirección (t_{ACC}) es igual al retardo desde $\overline{CE}$ hasta la salida (t_{CE}). Los datos están disponibles a la salida 120 ns (t_{OE}) después del flanco de bajada de $\overline{OE}$, suponiendo que $\overline{CE}$ era un nivel bajo y que las direcciones han permanecido estables durante un intervalo mínimo $t_{ACC} - t_{OE}$.

MODO DE RESERVA

La 2716 posee un modo de reserva que reduce la disipación de energía activa en un 75 %, de 525 mW a 132 mW. La 2716 se pone en modo de reserva aplicándole un nivel TTL alto a su entrada $\overline{CE}$. Cuando está en modo de reserva, las salidas se encuentran en el estado de alta impedancia, independiente de la entrada $\overline{OE}$.

SALIDA-INTERCONEXION OR

Como las 2716 se suelen utilizar en formaciones grandes de memoria, Intel ha previsto una función de control para 2 líneas que comprende este uso de conexiones a memorias múltiples. La función de control de dos líneas permite:

a) la disipación mínima posible de potencia en la memoria y
b) certeza absoluta de que no habrá contención a la salida del bus.

Para el uso más eficaz de estas dos líneas de control, se recomienda que $\overline{CE}$ (patilla 18) se decodifique y use como función principal de selección de dispositivo, mientras que $\overline{OE}$ (patilla 20) se haga conexión común a todos los dispositivos del array y se conecte a la línea de lectura (READ) desde el bus de control del sistema. Ello asegura que todos los dispositivos de memoria seleccionados estén en el modo de reserva de consumo bajo y que las patillas de salida estén sólo activas cuando se desea extraer datos de un dispositivo de memoria determinado.

PROGRAMACION

Al comienzo, después de cada borrado, todos los bits de la 2716 están en el estado «1». La información se introduce programando selectivamente «0» en las posiciones deseadas de bits. Aunque sólo se programan «ceros», ambos estado «1» y «0» pueden estar presentes en cada palabra de datos. La única manera de cambiar un «0» a un «1» es el borrado con luz ultravioleta.

La 2716 está en el modo de programación cuando la fuente de alimentación conectada a V_{PP} suministre 25 V y $\overline{OE}$ toma el valor V_{IH}. La información a programar se presenta de a 8 bits en paralelo en las patillas de salidas de datos. Los niveles necesarios para las direcciones y los datos de entrada son TTL.

Cuando la dirección y los datos se estabilizan se aplica un impulso TTL de programación de 50 m^2, activo de nivel alto, a la entrada $\overline{CE}$/PGM. Hay que aplicar un impulso de programación a cada una de las direcciones de las posiciones que se van a programar. Se puede programar cualquier dirección en cualquier momento, incluso individual, secuencial o aleatoriamente. El impulso de programación tendrá una anchura máxima de 55 ms. La 2716 no debe ser programada aplicando una señal de c.c. a la entrada $\overline{CE}$/PGM.

Se puede llevar a cabo fácilmente la programación múltiple de varias 2716 conectadas en paralelo, con la misma información, dada la sencillez de los requisitos de programación. Las entradas iguales de las 2716 conectadas en paralelo pueden unirse cuando se las programa con los mismos datos. La programación de todas las unidades 2716 conectadas en paralelo se consigue mediante un impulso de nivel alto TTL aplicado a las entradas $\overline{CE}$/PGM.

INHIBICION DEL PROGRAMA

La programación de varias 2716 en paralelo con datos distintos se lleva a cabo también fácilmente. Con la excepción de $\overline{CE}$/PGM, todas las entradas análogas ($\overline{OE}$ inclusive) de las 2716 conectadas en paralelo pueden ser comunes. Las 2716 que reciben en su entrada $\overline{CE}$/PGM un impulso de programación de nivel TTL junto con 25 V en V_{PP} resultan programadas. Las que reciben un nivel bajo en su entrada $\overline{CE}$/PGM quedan inhibidas y no reciben los datos.

VERIFICACION DEL PROGRAMA

Debe efectuarse la verificación de los bits programados para comprobar que presentan su estado lógico correcto. La verificación puede efectuarse con 25 V en V_{PP}. Excepto durante la programación y la verificación del programa, V_{PP} debe estar a 5 V.

Apéndice C3

intel

2102A, 2102AL/8102A-4*
MEMORIA RAM ESTATICA DE 1K×1 BIT

Modelo	Potencia en reserva (mW)	Potencia en funcionamiento (mW)	Acceso (ns)
2102AL-4	35	174	450
2102AL	35	174	350
2102AL-2	42	342	250
2102A-2	—	342	250
2102A	—	289	350
2102A-4	—	289	450

- **Tensión de alimentación única de +5 voltios.**
- **Compatible directamente con TTL: Todas las entradas y las salidas.**
- **Consumo reducido en el modo de reserva (2102AL).**
- **Salida en tres estados: posibilidad de nudo OR de interconexión.**
- **Entradas protegidas: Todas ellas disponen de protección contra la carga estática.**
- **Encapsulado barato: Configuración de 16 patillas en doble fila (DIL).**

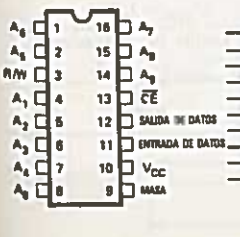
La Intel® 2102A es una memoria de acceso aleatorio de velocidad alta y de 1024 palabras de 1 bit, que usa dispositivos MOS canal-N integrados en un bloque monolítico. Usa una circuitería enteramente estable (estático) de c.c. y por ello no requiere ni reloj ni regeneración para funcionar. Los datos se leen no destructivamente y conservan la misma polaridad que los datos de entrada.

La 2102A está indicada como memoria en aquellas aplicaciones para las que son objetivos importantes de diseño un rendimiento elevado, coste bajo, gran capacidad de almacenamiento e interfaz sencillo. *Hay también disponible una versión (2102AL) de consumo reducido en reserva. Tiene las mismas características funcionales que la 2102A con la ventaja de una máxima disipación de potencia de 35 mW en reserva y 174 mW en trabajo.*

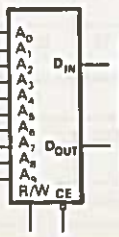
Es directamente compatible con TTL en todos los aspectos: entradas, salidas y tensión de alimentación única de +5 voltios. Una conexión separada para habilitación de chip (CE) permite la selección fácil e individual de una cualquiera cuando las salidas ofrecen un nudo de interconexión OR.

La Intel® 2102A está fabricada según el proceso de puertas de silicio canal-N, que permite el diseño y la producción de circuitos MOS de rendimiento elevado y uso fácil y suministra mayor densidad funcional para una patilla (chip) monolítica que la tecnología MOS convencional y puerta de silicio canal-P.

CONFIGURACION DE PATILLAS



SIMBOLO LOGICO



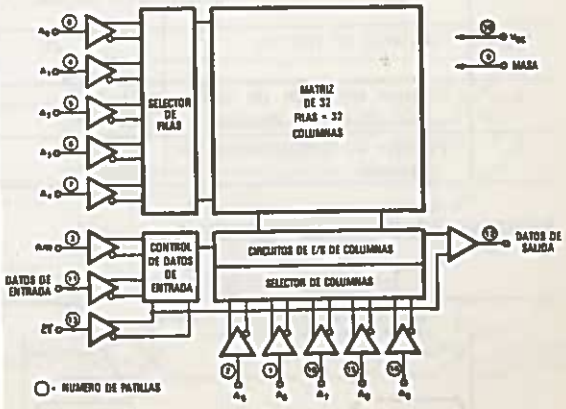
NOMBRE DE LAS PATILLAS

D_{IN}	ENTRADA DE DATOS
A_0-A_9	ENTRADAS DE DIRECCIONES
R/W	ENTR. DE LECTURA/ESCRITURA
$\overline{CE}$	HABILITACION DE PATILLA
D_{OUT}	SALIDA DE DATOS
V_{CC}	ALIMENTACION (+5V)

TABLA DE VERDA

$\overline{CE}$	R/W	D_{IN}	D_{OUT}	MODO
H	X	X	ALTA IMPEDANCIA (Z)	NO SELECCIONADA
L	L	L	L	ESCRIBIR 0
L	L	H	H	ESCRIBIR 1
L	H	X	D_{OUT}	LEER

DIAGRAMA DE BLOQUES



*Todas las características de la 8102A son idénticas a las de la 2102A-4.

Impreso con permiso de Intel Corporation © 1978

REGIMENES MAXIMOS ABSOLUTOS*

Temperatura ambiente baja polarización	−10 °C a 80 °C
Temperatura de almacenamiento	−65 °C a +150 °C
Tensión en cualquier patilla con respecto a tierra	−0,5 V a +7 V
Disipación de potencia	1 W

*COMENTARIO: Los excesos por encima de los valores indicados bajo los «regímenes máximos absolutos» pueden ocasionar daños permanentes al dispositivo. Esto es un régimen de exceso solamente y el funcionamiento del dispositivo no está implicado bajo estas u otras condiciones por encima de las indicadas en el apartado operativo de estas especificaciones. La exposición durante largos períodos a las condiciones de los regímenes máximos absolutos puede afectar a la fiabilidad del dispositivo.

CARACTERISTICAS DE FUNCIONAMIENTO Y DE C.C.

T_A=0°C a 70°C, V_{CC}=5 V±5 %, salvo que se indique otra cosa.

Símbolo	Parámetro	2102A, 2102A-4 2102AL, 2102AL-4 Límites Mín.	Típ.[1]	Máx.	2102A-2, 2102AL-2 Límites Mín.	Típ.[1]	Máx.	Unidad	Condiciones de prueba
I_{LI}	Corriente de carga a la entrada		1	10		1	10	μA	V_{IN}=0 a 5,25 V
I_{LOH}	Corriente de fuga a la salida		1	5		1	5	μA	$\overline{CE}$=2,0 V, $V_{OUT}=V_{OH}$
I_{LOL}	Corriente de fuga a la salida		−1	−10		−1	−10	μA	$\overline{CE}$=2,0 V, V_{OUT}=0,4 V
I_{CC}	Corriente de alimentación		33	Nota 2		45	65	mA	Todas las entradas =5,25 V Salida de datos abierta, T_A=0 °C
V_{IL}	Tensión de entrada para nivel bajo	−0,5		0,8	−0,5		0,8	V	
V_{IH}	Tensión de entrada para nivel alto	2,0		V_{CC}	2,0		V_{CC}	V	
V_{OL}	Tensión de salida para nivel bajo			0,4			0,4	V	I_{OL}=2,1 mA
V_{OH}	Tensión de salida para nivel alto	2,4			2,4			V	I_{OH}=−100 μA

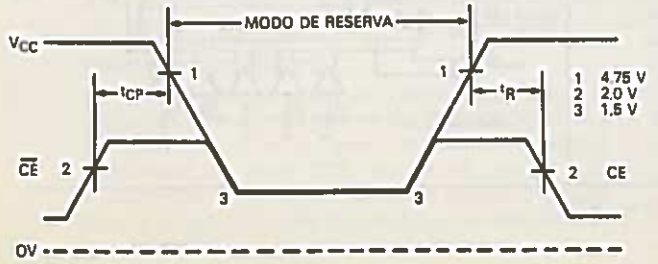
Notas: 1. Los valores típicos para T_A=25 °C y tensión de alimentación nominal.
2. El valor máximo de I_{CC} es 55 mA para las 2102A y 2102A-4 y 33 mA para las 2102AL y 2102AL-4.

CARACTERISTICAS EN RESERVA 2102AL, 2102AL-2 y 2102AL-4.

(Disponibles sólo en encapsulado plástico.) TA=0°C a 70°C

Símbolo	Parámetro	2102AL, 2102AL-4 Límites Mín.	Típ.[1]	Máx.	2102AL-2 Límites Mín.	Típ.[1]	Máx.	Unidad	Condiciones de prueba
V_{PD}	V_{CC} en reserva	1,5			1,5			V	
V_{CES}[3]	Polarización $\overline{CE}$ en reserva	2,0			2,0			V	2,0 V≤V_{PD}≤V_{CC} Máx
		V_{PD}			V_{PD}			V	1,5 V≤V_{PD}<2,0 V
I_{PD1}	Corriente en reserva		15	23		20	28	mA	Todas las entradas V_{PD1}=1,5 V
I_{PD2}	Corriente en reserva		20	30		25	38	mA	Todas las entradas V_{PD2}=2,0 V
t_{CP}	Tiempo entre fin de selección de chip y reserva	0			0			ns	
T_R[3]	Tiempo de recuperación en reserva	t_{RC}			t_{RC}			ns	

FORMAS DE ONDA EN RESERVA



NOTAS:
1. Los valores típicos para T_A=25 °C.
2. Considérense las condiciones de prueba como se indica: Si la tensión de reserva (V_{PD}) queda entre 5,25 V (V_{CC} máx.) y 2,0 V, entonces $\overline{CE}$ ha de mantenerse a 2,0 V mín. (V_{IH}). Si la tensión de reserva es menor de 2,0 V, pero mayor que 1,5 V (V_{PD} mín.), entonces, $\overline{CE}$ y la tensión de reserva deben ser como mínimo del mismo valor 0; si son diferentes, $\overline{CE}$ debe ser el más positivo de ambos.
3. t_R=t_{RC} (Tiempo del ciclo de lectura).

CARACTERISTICAS EN C.A. T_A=0°C a 70°C, V_{CC}=5 V± %, salvo que se indique otra cosa.

CICLO DE LECTURA

Símbolo	Parámetro	2102A-2, 2102AL-2 Límites (ns) Mín.	Máx.	2102A, 2102AL Límites (ns) Mín.	Máx.	2102A-4, 2102AL-4 Límites (ns) Mín.	Máx.
t_{RC}	Ciclo de lectura	250		350		450	
t_A	Tiempo de acceso		250		350		450
t_{CO}	Tiempo entre habilitación de chip y salida		130		180		230
t_{OH1}	Datos de lectura válidos previos respecto de dirección	40		40		40	
t_{OH2}	Datos de lectura válidos previos respecto de habilitación de patilla	0		0		0	

CICLO DE ESCRITURA

t_{WC}	Ciclo de escritura	250	350	450
t_{AW}	Tiempo desde dirección hasta listo para escribir	20	20	20
t_{WP}	Anchura del impulso de escritura	180	250	300
t_{WR}	Tiempo de recuperación de escritura	0	0	0
t_{DW}	Tiempo de preparación de datos	180	250	300
t_{DH}	Tiempo de retención de datos	0	0	0
t_{CW}	Tiempo entre habilitación de chip y habilitación de escritura	180	250	300

CONDICIONES EN C.A. PARA LA PRUEBA

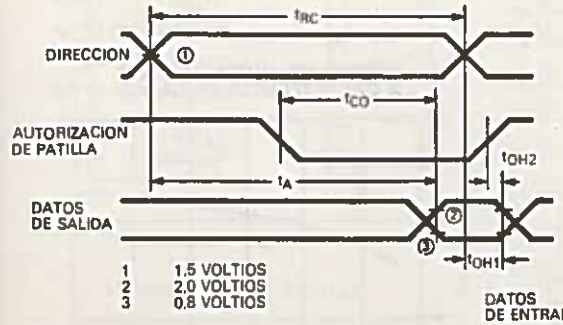
Niveles del impulso de entrada:		0,8 V a 2,0 V
Tiempos de subida y bajada a la entrada:		10 ns
Niveles de referencia para medida de tiempos	Entradas:	1,5 voltios
	Salida:	0,8 y 2,0 voltios
Carga a la salida:		1 puerta TTL y C_L=100 pF

CAPACITANCIA[2] T_A=25°C, f=1 MHz

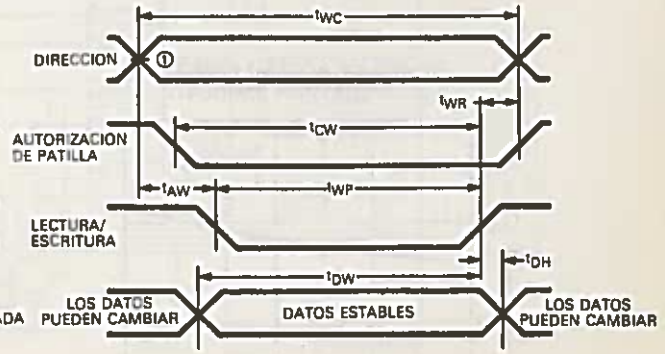
SIMBOLO	PRUEBA	LIMITES (pF) Tip.[1]	Máx.
C_{IN}	CAPACITANCIA DE ENTRADA (TODAS LAS PATILLAS DE ENTRADA) V_{IN}=0V	3	5
C_{OUT}	CAPACITANCIA DE SALIDA V_{OUT}=0V	7	10

FORMAS DE ONDA

CICLO DE LECTURA



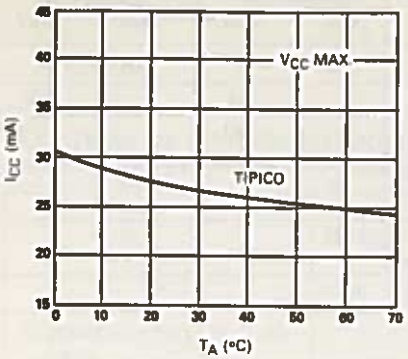
CICLO DE ESCRITURA



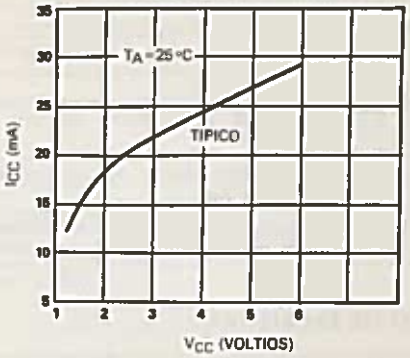
NOTAS:
1. Los valores típicos son para T_A=25°C y tensión de alimentación nominal.
2. Este parámetro se muestra periódicamente y se no prueba al 100 %.

CARACTERISTICAS TIPICAS EN C.C. Y C.A.

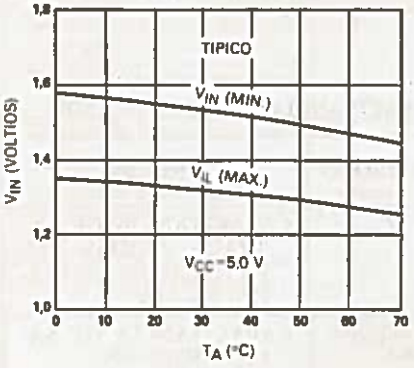
CORRIENTE DE ALIMENTACION
FRENTE A TEMPERATURA AMBIENTE



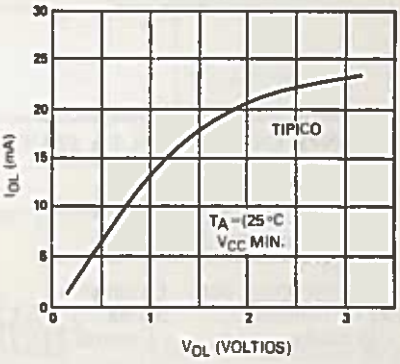
CORRIENTE DE ALIMENTACION
FRENTE A TENSION DE ALIMENTACION



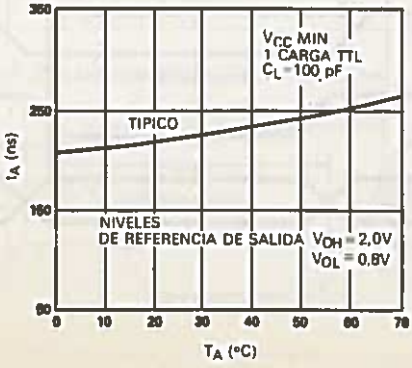
LIMITES DE V_{IN} FRENTE
A TEMPERATURA



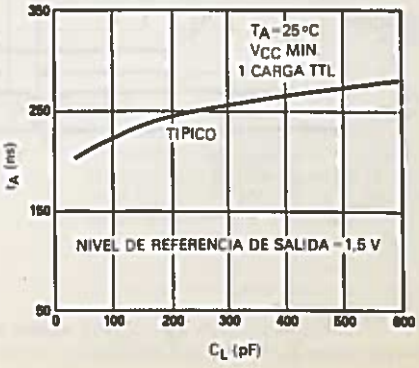
CORRIENTE DE RETORNO
A LA SALIDA FRENTE
A TENSION DE SALIDA



TIEMPO DE ACCESO FRENTE
A TEMPERATURA AMBIENTE



TIEMPO DE ACCESO FRENTE
A CAPACITANCIA DE CARGA



Apéndice C4

intel

2114A
MEMORIA RAM ESTATICA DE 1024 × 4 BITS

	2114AL-1	2114AL-2	2114AL-3	2114AL-4	2114A-4	2114A-5
Tiempo máximo de acceso (ns)	100	120	150	200	200	250
Corriente máxima (mA)	40	40	40	40	70	70

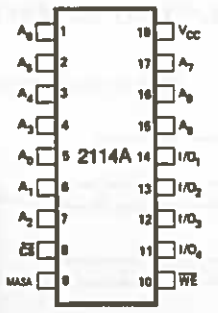
- **Tecnología HMOS**
- **Baja disipación, alta velocidad.**
- **Tiempo de acceso idéntico a la duración del ciclo.**
- **Alimentación única a +5 V ± 10 %.**
- **Encapsulado de alta densidad con 18 patillas.**
- **Memoria enteramente estática sin requerir reloj ni selección en tiempo.**
- **Compatible directamente con TTL: Todas las entradas y salidas.**
- **Entrada y salida de datos en común mediante las salidas de tres estados.**
- **Modelo 2114 perfeccionado.**

La Intel® 2114A es una memoria de acceso aleatorio, estática, de 4096 bits organizada como 1024 palabra de 4 bits y que usa HMOS, una tecnología MOS de alto rendimiento. Utiliza una circuiteria enteramente estable (estática), tanto en la matriz como en el decodificador, por lo que no precisa reloj ni regeneración para funcionar. El acceso de los datos es especialmente sencillo, ya que no se necesita esperar los tiempos de preparación de direcciones. Los datos se leen no destructivamente y presentan la misma polaridad que los de entrada. Se han previsto algunas patillas comunes para la entrada y la salida.

La 2114A está indicada como memoria en aquellas aplicaciones para las que son objetivos importantes de diseño un rendimiento elevado, la alta fiabilidad de los HMOS, coste bajo, gran capacidad de almacenamiento e interfaz sencillo. La 2114A se presenta en cápsulas de 18 patillas para preservar la mayor densidad posible.

Es compatible directamente con TTL en todos los aspectos: entradas, salidas y tensión de alimentación única de +5 V. Una conexión separada de selección de chip (CS) permite la selección fácil de una patilla individual cuando las salidas ofrecen un nudo de interconexión OR.

CONFIGURACION DE PATILLAS



SIMBOLO LOGICO

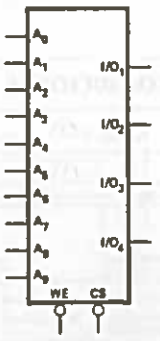
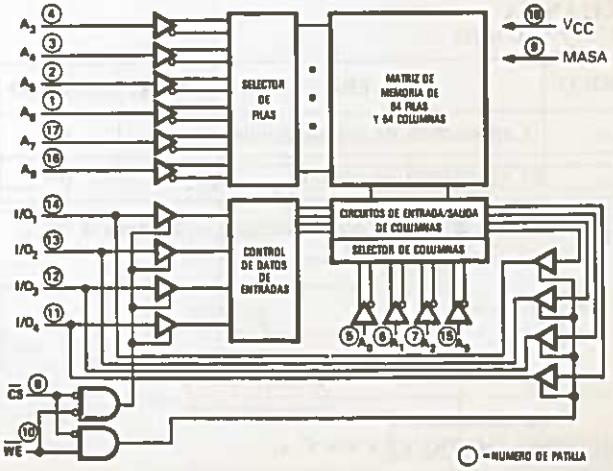


DIAGRAMA DE BLOQUES



NOMBRE DE LAS PATILLAS

A_0–A_9	ENTRADAS DE DIRECCIONES	V_{cc}	ALIMENTACION (+5V)
$\overline{WE}$	HABILITACION DE ESCRITURA	GND	MASA
$\overline{CS}$	SELECCION DE CHIP		
I/O_1–I/O_4	ENTRADA/SALIDA DE DATOS		

Impreso con permiso de Intel Corporation © 1980.

REGIMENES MAXIMOS ABSOLUTOS*

Temperatura ambiente baja polarización	−10 °C a 80 °C
Temperatura de almacenamiento	−65 °C a 150 °C
Tensión en cualquier patilla con respecto a tierra	−3,5 a +7 V
Disipación de potencia	1,0 W
Corriente de salida en continua	5 mA

*COMENTARIO: Los excesos por encima de los valores indicados bajo los «regímenes máximos absolutos» pueden ocasionar daños permanentes al dispositivo. Esto es un régimen de exceso solamente y el funcionamiento del dispositivo no está implicado bajo estas u otras condiciones por encima de las indicadas en el apartado operativo de estas especificaciones. La exposición durante largos períodos a las condiciones de los regímenes máximos absolutos puede afectar a la fiabilidad del dispositivo.

CARACTERISTICAS DE FUNCIONAMIENTO Y C.C.

T_A=0 °C a 70 °C, V_{CC}=5 V±10 %, salvo que se indique otra cosa.

SIMBOLO	PARAMETRO	2114AL-1/L-2/L-3/L-4 Mín.	Típ.[1]	Máx.	2114A-4/-5 Mín.	Típ.[1]	Máx.	UNIDAD	CONDICIONES
I_{LI}	Corriente de carga a la entrada (en todas las patillas de entrada)			10			10	µA	V_{IN}=0 a 5,5 V
$\|I_{LO}\|$	Corriente de fuga de entrada/salida			10			10	µA	$\overline{CS}=V_{IH}$ $V_{I/O}$= MASA a VCC
I_{CC}	Corriente de alimentación		25	40		50	70	mA	V_{CC}=máx., $I_{I/O}$=0 mA, T_A=0 °C
V_{IL}	Voltaje de entrada para nivel bajo	−3,0		0,8	−3,0		0,8	V	
V_{IH}	Voltaje de entrada para nivel alto	2,0		6,0	2,0		6,0	V	
I_{OL}	Corriente de salida para nivel bajo	2,1	9,0		2,1	9,0		mA	V_{OL}=0,4 V
I_{OH}	Corriente de salida para nivel alto	−1,0	−2,5		−1,0	−2,5		mA	V_{OH}=2,4 V
I_{OS}[2]	Corriente de salida en cortocircuito			10			40	mA	

NOTAS:
1. Los valores típicos son para T_A=25 °C y V_{CC}=5,0 V.
2. La duración no excede a 30 segundos.

CAPACITANCIA

T_A=25 °C, f=1,0 MHz

SIMBOLO	PRUEBA	MAX.	UNIDAD	CONDICIONES
$C_{I/O}$	Capacitancia de entrada/salida	5	pF	V_{IO}=OV
C_{IN}	Capacitancia de entrada	5	pF	V_{IN}=OV

NOTA: Este parámetro se muestrea periódicamente y no se prueba al 100 %.

CONDICIONES DE PRUEBA EN C.A.

Niveles del impulso de entrada	0,8 V a 2,0 V
Tiempo de subida y bajada a la entrada	10 ns
Niveles de tiempo a la entrada y salida	1,5 V
Carga de la salida	1 puerta TTL y C_L=100 pF

CARACTERISTICAS EN C.A. T_A=0°C a 70°C, V_{CC}+5 V±10 %, salvo que se indique otra cosa.

CICLO DE LECTURA[1]

SIMB.	PARAMETRO	2114AL-1 Mín.	2114AL-1 Máx.	2114AL-2 Mín.	2114AL-2 Máx.	2114AL-3 Mín.	2114AL-3 Máx.	2114A-4/L-4 Mín.	2114A-4/L-4 Máx.	2114A-5 Mín.	2114A-5 Máx.	UNI.
t_{RC}	Duración del ciclo de lectura	100		120		150		200		250		ns
t_A	Tiempo de acceso		100		120		150		200		250	ns
t_{CO}	Selección de chip hasta salida válida		70		70		70		70		85	ns
t_{CX}	Selección de chip hasta salida activa	10		10		10		10		10		ns
t_{OTD}	Salida triestado desde fin de selección		30		35		40		50		60	ns
t_{OHA}	Salida retenida desde cambio de dirección	15		15		15		15		15		ns

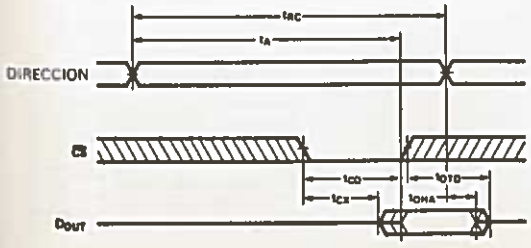
CICLO DE ESCRITURA[2]

SIMB.	PARAMETRO	2114AL-1 Mín.	2114AL-1 Máx.	2114AL-2 Mín.	2114AL-2 Máx.	2114AL-3 Mín.	2114AL-3 Máx.	2114A-4/L-4 Mín.	2114A-4/L-4 Máx.	2114A-5 Mín.	2114A-5 Máx.	UNI.
t_{WC}	Duración del ciclo de escritura	100		120		150		200		250		ns
t_W	Tiempo de escritura	75		75		90		120		135		ns
t_{WR}	Tiempo de liberación de escritura	0		0		0		0		0		ns
t_{OTW}	Salida triestado desde escritura		30		35		40		50		60	ns
t_{DW}	Datos hasta solape tiempo de escritura	70		70		90		120		135		ns
t_{DH}	Datos retenidos desde tiempo de escritura	0		0		0		0		0		ns

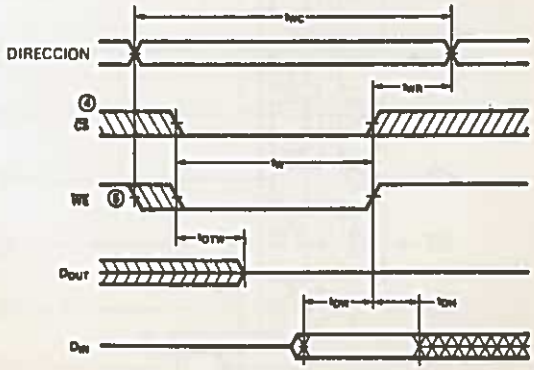
NOTAS:
1. El impulso de lectura ocurre durante el solape de $\overline{CS}$ a nivel bajo y $\overline{WE}$ a nivel alto.
2. El impulso de escritura ocurre mientras que $\overline{CS}$ está a nivel bajo y $\overline{WE}$ también. t_W se mide desde el último $\overline{CS}$ o $\overline{WE}$ que se hizo nivel bajo hasta el primer $\overline{CS}$ o $\overline{WE}$ que se hace alto.

FORMAS DE ONDA

CICLO DE LECTURA[3]



CICLO DE ESCRITURA



NOTAS:
3. $\overline{WE}$ está a nivel alto durante el ciclo de lectura.
4. Si la transición de $\overline{CS}$ a nivel bajo ocurre simultáneamente con la transición de $\overline{WE}$ a nivel bajo también, los buffers de salida permanecen en el estado de alta impedancia.
5. $\overline{WE}$ debe estar a nivel alto durante todas las transiciones de dirección.

Apéndice C5

intel

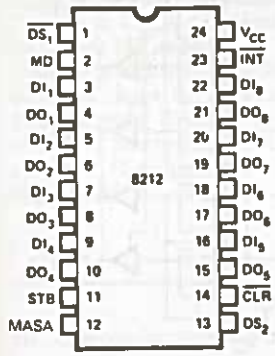
8212
PORT (PUERTO) DE ENTRADA/SALIDA DE 8 BITS

- **Registro de datos para 8 bits totalmente en paralelo y memoria intermedia.**
- **Flip-flop para petición de servicio en la generación de interrupciones.**
- **Corriente de carga de entrada reducida −0,25 mA máx.**
- **Salidas triestados.**
- **Corriente de retorno a la salida de 15 mA.**
- **Tensión de salida de 3,65 V para nivel alto, para interfaz directo con las CPUs 8008, 8080 A o 8085A.**
- **Puesta a cero asíncrona del registro.**
- **Sustituciones de memoria intermedias, circuitos de enclavamiento («latch») y multiplexores en los sistemas de microcomputadoras.**
- **Reduce la cuenta de encapsulados en cada sistema.**

El port 8212 de entrada/salida consiste en un «latch» para 8 bits con buffers de salida triestado junto con la lógica correspondiente de control y selección de dispositivo. También se incluye un flip-flop para la solicitud de servicio cuando hay que generar y controlar interrupciones al microprocesador.

El dispositivo es de naturaleza multimodal. Se puede emplear para formar «latchs», «buffers» controlados por puertas o multiplexores. Por consiguiente, con este dispositivo se pueden llevar a cabo todas las funciones principales periféricas y de entrada/salida de una red de microcomputadora.

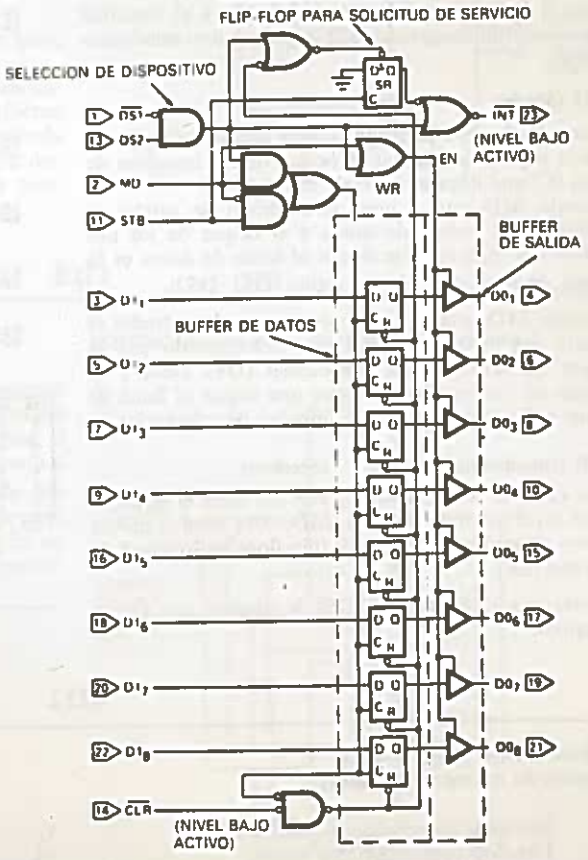
CONFIGURACION DE PATILLAS



NOMBRE DE LAS PATILLAS

DI_1 DI_8	ENTRADA DE DATOS
DO_1 DO_8	SALIDA DE DATOS
$\overline{DS_1}$, DS_2	SELECTOR DE DISPOSITIVOS
MD	MODO
STB	IMPULSO DE REFERENCIA (STROBE)
$\overline{INT}$	INTERRUPCION (NIVEL BAJO ACTIVO)
$\overline{CLR}$	PUESTA A CERO (NIVEL BAJO ACTIVO)

DIAGRAMA LOGICO



Impreso con permiso de Intel Corporation © 1980.

DESCRIPCION FUNCIONAL

Latch de datos

Los 8 flip-flop que componen el latch de datos son del tipo «D» de retardo. La salida (Q) del biestable (flip-flop) sigue a la entrada de datos mientras que la entrada de reloj (C) está a nivel alto. El enclavamiento («latching») ocurre cuando el reloj retorna al nivel bajo.

Los datos así enclavados en los biestables se borran mediante la entrada asíncrona para puesta a cero ($\overline{CLR}$). [Nota: El reloj (C) anula la puesta a cero ($\overline{CLR}$).]

Buffer de salida

Las salidas del «latch» de datos (Q) están conectadas a buffers de salida, de 3 estados, no inversores. Estos buffers poseen una línea común de control (EN); esta línea de control permite que el buffer tramite los datos de la salida del latch (Q), o bien inhabilita al buffer, forzando su salida al estado de alta impedancia (triestado).

El estado de alta impedancia permite al diseñador conectar directamente la 8212 al bus de datos bidireccional del microprocesador.

Lógica de control

La 8212 posee las entradas de control $\overline{DS1}$, DS2, MD y STB. Estas entradas se usan para controlar la selección del dispositivo, el latch de datos, el estado de los buffers de salida y el biestable (flip-flop) para solicitud de servicio.

$\overline{DS1}$, DS2 (selección de dispositivo)

Estas dos entradas sirven para la selección del dispositivo. Cuando $\overline{DS1}$ está a nivel bajo y DS2 a nivel alto ($\overline{DS1} \cdot DS2$), el dispositivo está seleccionado. En este estado el buffer de salida está habilitado y el biestable para solicitud de servicio (SR) se pone a uno asíncronamente.

MD (Modo)

Esta entrada sirve para controlar el estado del buffer de salida y para determinar el origen de los impulsos de reloj (C) que llega al latch de datos.
Cuando MD está a nivel alto (modo de salida) se habilitan los buffers de salida y el origen de los impulsos del reloj (C) que llegan al latch de datos es la lógica de selección del dispositivo ($\overline{DS1} \cdot DS2$).

Cuando MD está a nivel bajo (modo de entrada) el estado del buffer de salida está determinado por la lógica de selección del dispositivo ($\overline{DS1} \cdot DS2$) y el origen de los impulsos de reloj que llegan al latch de datos es la entrada STB (de impulso de referencia).

STB (Impulso de referencia - «Strobe»)

Esta entrada se toma como reloj (C) para el latch de datos en el modo de entrada (MD = 0) y para la puesta a cero asíncrona del biestable (flip-flop) de solicitud de servicio (SR).

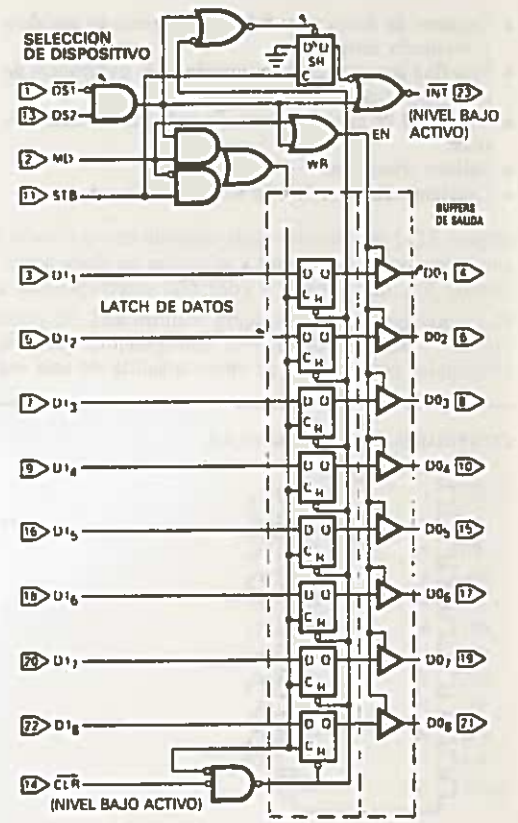
Nótese que el biestable SR es de disparo con flanco negativo.

Flip-flop para solicitud de servicio

Este biestable (SR) se usa para generar y controlar interrupciones en los sistemas de microcomputadoras. Se pone a uno asíncronamente mediante la entrada $\overline{CLR}$ (nivel bajo activo). Cuando el biestable (SR) esta puesto a uno indica estado de no interrupción.

La salida (Q) del biestable (SR) está conectada a la entrada inversora de una puerta «NOR». La otra entrada de la «NOR» no es inversora y queda conectada a la lógica de selección del dispositivo ($\overline{DS1} \cdot DS2$). La salida de la puerta NOR ($\overline{INT}$) es activa en nivel bajo (estado de interrupción) para ser conectada a la entrada de nivel bajo activo de los circuitos de generación de prioridad.

FLIP-FLOP DE SOLICITUD DE SERVICIO



STB	MD	DS_1 DS_2	IGUALES QUE LOS DATOS DE SALIDA
0	0	0	TRIESTADO
1	0	0	TRIESTADO
0	1	0	LATCH DE DATOS
1	1	0	LATCH DE DATOS
0	0	1	LATCH DE DATOS
1	0	1	ENTRADA DE DATOS
0	1	1	ENTRADA DE DATOS
1	1	1	ENTRADA DE DATOS

$\overline{CLR}$ — PONE A CERO EL LATCH DE DATOS
PONE A UNO EL BIESTABLE (SR)
(NO AFECTA AL BUFFER DE SALIDA)

CLR	$\overline{DS_1}$ DS_2	STB	*SR	INT
0	0	0	1	1
0	1	0	0	0
1	1	↘	0	0
1	1	0	1	0
1	0	0	1	1
1	1	↘	1	0

* BIESTABLE SR INTERNO

8212

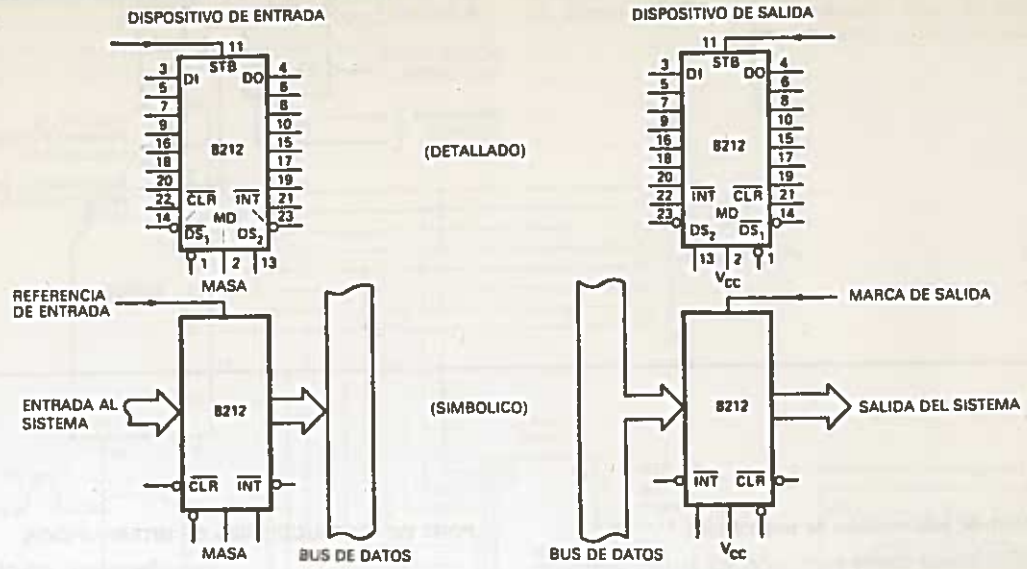
Aplicaciones de la 8212 en los sistemas de microcomputadoras

I. Símbolos esquemáticos básicos.
II. Flip-flop controlado por puerta.
III. Controlador para bus bidireccional.
IV. Port de entrada para interrupciones.
V. Port de instrucciones de interrupción.
VI. Port de salida.
VII. Latch de estado para 8080A.
VIII. Latch de direcciones para 8085A

I. Símbolos esquemáticos básicos

Damos dos ejemplos de las maneras de dibujar la 8212 en los esquemas de una instalación: (1) la superior es una vista detallada que muestra los números de las patillas y la (2) debajo es la vista simbólica que muestra la entrada o salida del sistema como si se tratara de un bus (un bus que contiene 8 líneas en paralelo). La salida dal bus de datos referencia simbólicamente las 8 líneas en paralelo.

SIMBOLOS ESQUEMATICOS BASICOS



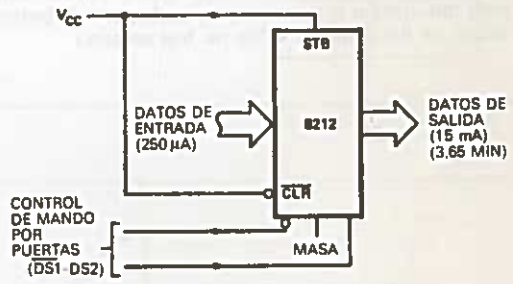
II. Buffer controlado por puertas (triestado)

El uso más sencillo de la 8212 es el de buffer controlado por puertas. Uniendo la señal de modo a nivel bajo con la entrada de referencia a nivel alto, el latch de datos se comporta como una puerta de paso directo. Los buffers de salida se habilitan entonces mediante la lógica de selección del dispositivo $\overline{DS1}$ y DS2.

Cuando la lógica de selección del dispositivo es falsa, las salidas son triestados.

Cuando la lógica de selección del dispositivo es verdadera, los datos de entrada al sistema se transfieren directamente a la salida. La carga de datos de entrada es de 250 microamperios. Los datos de salida pueden consumir 15 miliamperios. La salida mínima de nivel alto es 3,65 V.

BUFFER CONTROLADO POR PUERTAS

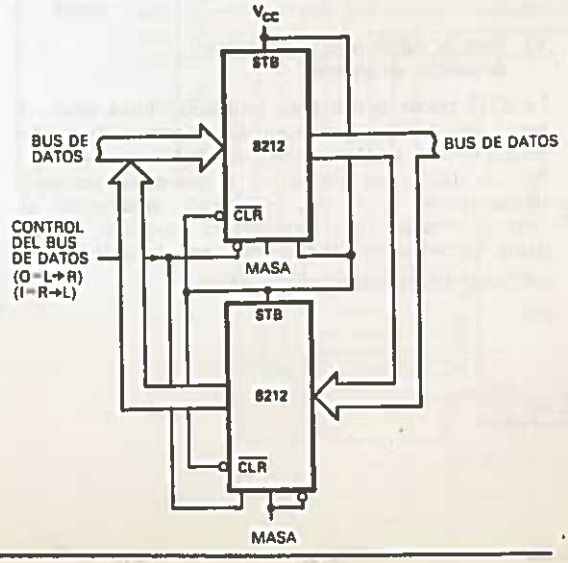


8212

III. Controlador para bus bidireccional

Se puede usar un par de 8212 conectadas en oposición como controlador para bus bidireccional de excitación simétrica. Los dispositivos están gobernados por el control de entrada del bus de datos que se conecta a $\overline{DS1}$ en la primera 8212 y a DS2 en la segunda. Un dispositivo es activo y funciona como un buffer de paso directo y el otro queda en el modo triestado. Es un circuito muy útil en el diseño de los sistemas pequeños.

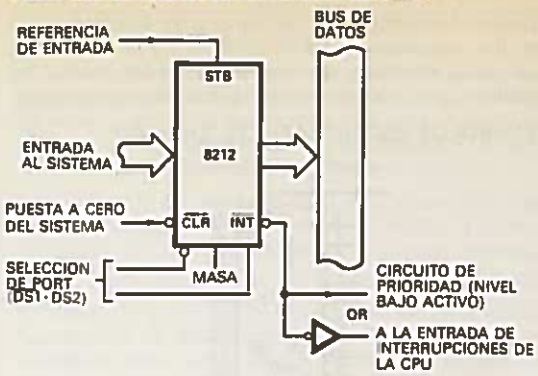
CONTROLADOR PARA BUS BIDIRECCIONAL



IV. Port de entrada de interrupciones

La utilidad de una 2812 es la de un port de entrada al sistema que acepta una referencia de su fuente de entrada, que a su vez pone a cero al biestable de solicitud de servicio e interrumpe al procesador. Este pasa entonces a través de la rutina de servicio, identifica el port y hace que la lógica de selección del dispositivo adopte su valor verdadero (habilitando la entrada de datos del sistema desde el bus).

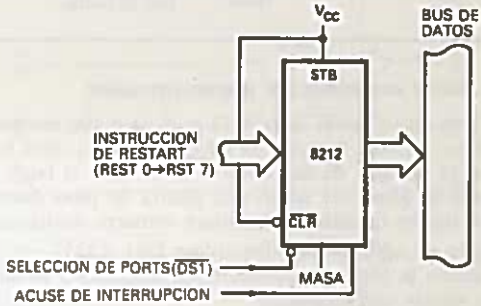
PORT DE ENTRADA DE INTERRUPCIONES



V. Port de instrucciones de interrupción

La 8212 puede usarse para gobernar la instrucción de interrupción, que suele ser las instrucciones RESTART, en el bus de datos. El dispositivo se habilita con la señal de acuse de interrupción procedente del microprocesador y con la de la selección de port. Esta señal está normalmente conectada a tierra. ($\overline{DS1}$ puede usarse para multiplexar una gran variedad de ports de instrucciones de interrupción sobre un bus común.)

PORT DE INSTRUCCIONES DE INTERRUPCION

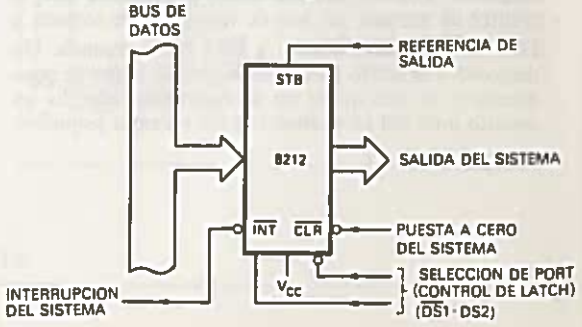


8212

VI. Port de salida (con intercambio de señales de control)

La 8212 puede usarse para transmitir datos desde el bus a una salida del sistema. La referencia de salida podría ser una señal para intercambio tal como «recepción de datos» del dispositivo al que el sistema está transmitiendo. A su vez, éste puede interrumpir al microprocesador para significar la recepción de los datos. La selección del port proviene de la lógica de selección del dispositivo (DS1 · DS2).

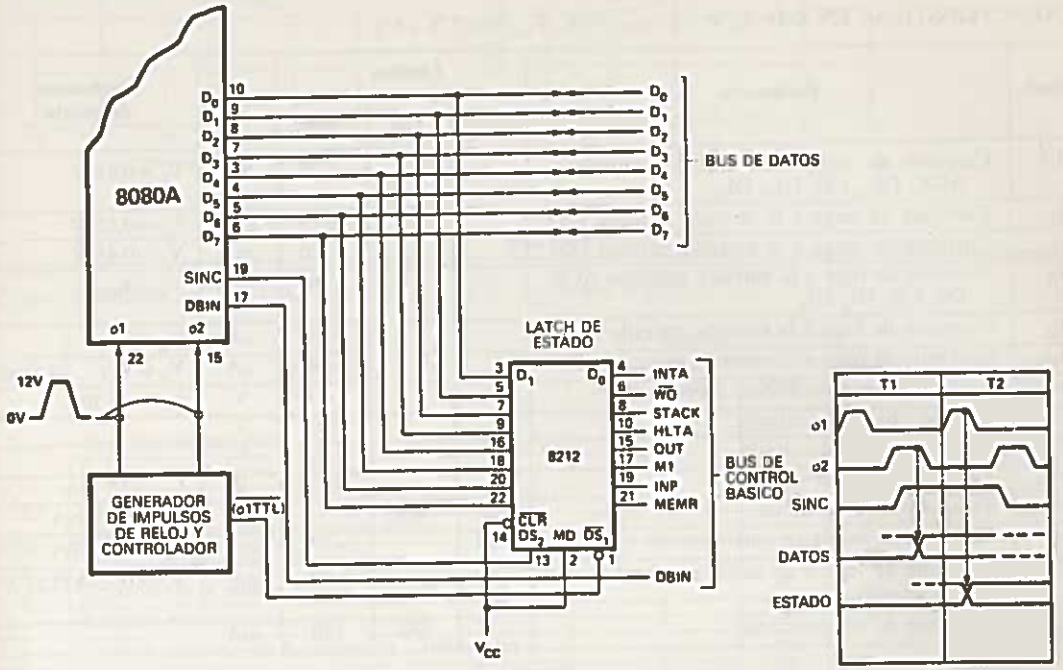
PORT DE SALIDA (CON INTERCAMBIO DE SEÑALES DE CONTROL)



VII. Latch de estado para 8080A

Aquí se usa la 8212 como latch de estado para una microcomputadora 8080A. La entrada al latch 8212 viene directamente del bus de datos del 8080A. El cronograma muestra que cuando la señal de sincronismo SYNC toma su valor verdadero, estando conectada a la entrada DS2 y cuando la señal fase 1 toma también su valor verdadero, que es un nivel TTL procedente del generador de impulsos de reloj, entonces los datos de estado se enclavan en la 8212.

Nota: La señal de modo se mantiene a nivel alto de manera que la salida del latch queda activada y autorizada todo el tiempo.
Queda señalado que las dos zonas de interés son el bus bidireccional de datos del microprocesador y el bus de control.

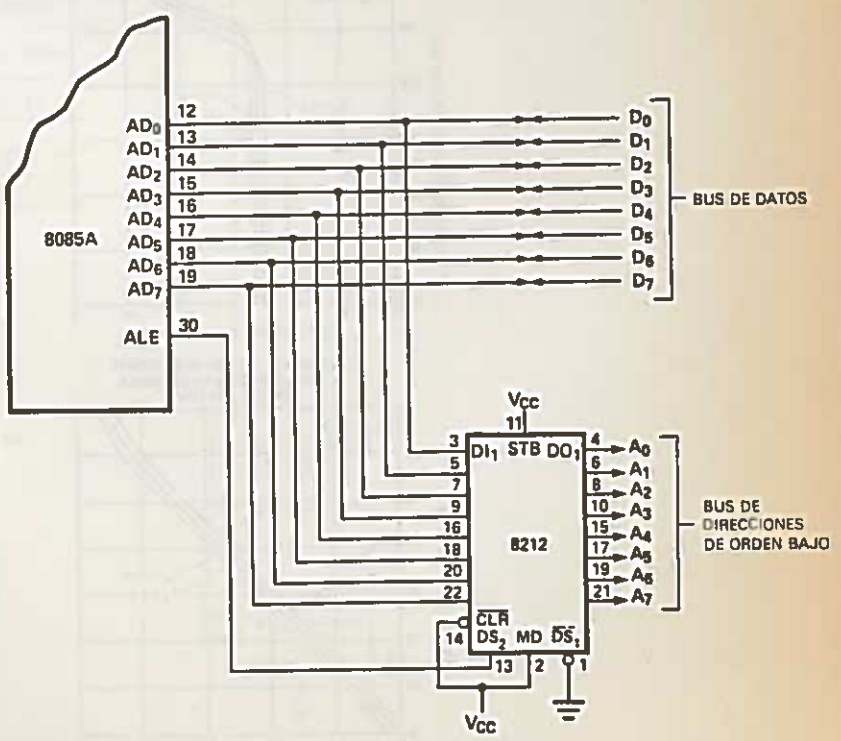


8212

VIII. Latch de direcciones de orden bajo para el 8085A

El microprocesador 8085A usa un bus multiplexado de datos/direcciones que contiene los 8 bits menos significativos de la información de la dirección durante la primera parte del ciclo de máquina. El mismo bus contiene datos algún tiempo después dentro del mismo ciclo. El 8085 genera una señal de habilitación de enclavamiento de direcciones (ALE) para ser usada por la señal 8212 para enclavar la dirección de modo que esté disponible durante el ciclo de máquina completo.

Nota: En esta configuración la entrada MODE se conecta a nivel alto, de modo que los buffers de salida de la 8212 se mantengan activos en todo momento.



REGIMENES MAXIMOS ABSOLUTOS*

Temperatura bajo polarización en plástico	0°C a +70°C
Temperatura de almacenamiento	−65°C a +160°C
Tensión en todas las salidas o en la alimentación	0,5 a +7 voltios
Tensión en todas las entradas	−1,0 a 5,5 voltios
Corrientes de salida	100 mA

*COMENTARIO: Los excesos por encima de los valores indicados bajo los «regímenes máximos absolutos» pueden ocasionar daños permanentes al dispositivo. Esto es un régimen de exceso solamente y el funcionamiento del dispositivo no está implicado bajo estas u otras condiciones por encima de las indicadas en el apartado operativo de estas especificaciones. La exposición durante largos períodos a las condiciones de los regímenes máximos absolutos puede afectar a la fiabilidad del dispositivo.

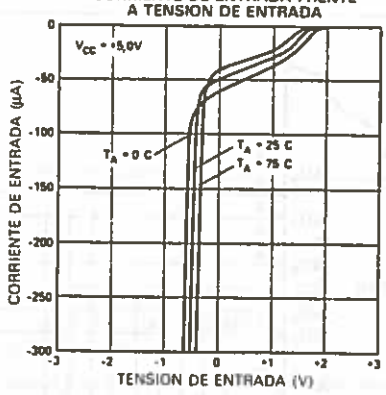
CARACTERISTICAS EN C.C. $T_A = 0°C$ a $+75°C$, $V_{CC} = +5\ V \pm 5\%$.

Símb.	Parámetro	Límites			Unidad	Condiciones de prueba
		Mín.	Tip.	Máx.		
I_F	Corriente de carga a la entrada, entradas ACK, DS_2, CR, DI_1, DI_8			−0,25	mA	$V_F = 0{,}45$ V
I_F	Corriente de carga a la entrada, entrada MD			−0,75	mA	$V_F = 0{,}45$ V
I_F	Corriente de carga a la entrada, entrada DS1			−1,0	mA	$V_F = 0{,}45$ V
I_R	Corriente de fuga a la entrada, entradas ACK, DS, CR, DI_1-DI_8			10	μA	$V_R \leq V_{CC}$
I_R	Corriente de fuga a la entrada, entrada MD			30	μA	$V_R \leq V_{CC}$
I_R	Corriente de fuga a la entrada, entrada DS1			40	μA	$V_R \leq V_{CC}$
V_C	Fijación de tensión directa de entrada			−1	V	$I_C = -5$ mA
V_{IL}	Tensión «baja» de entrada			0,85	V	
V_{IH}	Tensión «alta» de entrada	2,0			V	
V_{OL}	Tensión «baja» de salida			0,45	V	$I_{OL} = 15$ mA
V_{OH}	Tensión «alta» de salida	3,65	4,0		V	$I_{OH} = -1$ mA
I_{SC}	Corriente de salida en cortocircuito	−15		−75	mA	$V_O = 0$ V, $V_{CC} = 5$ V
$\|I_O\|$	Corriente de fuga a la salida en el estado de alta impedancia			20	μA	$V_O = 0{,}45$ V/5,25 V
I_{CC}	Corriente de alimentación		90	130	mA	

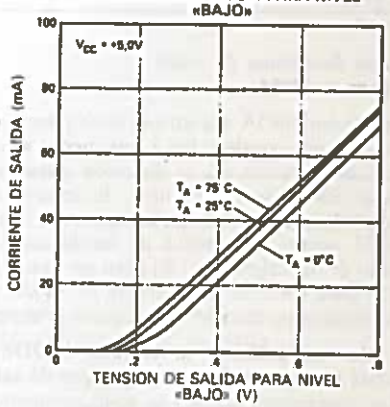
8212

CARACTERISTICAS TIPICAS

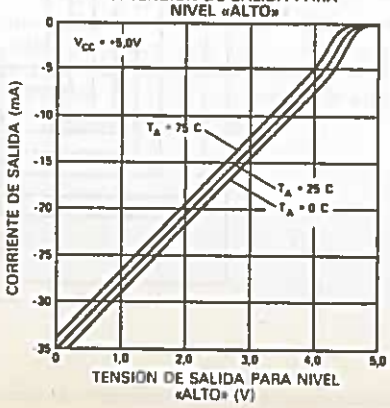
CORRIENTE DE ENTRADA FRENTE A TENSION DE ENTRADA



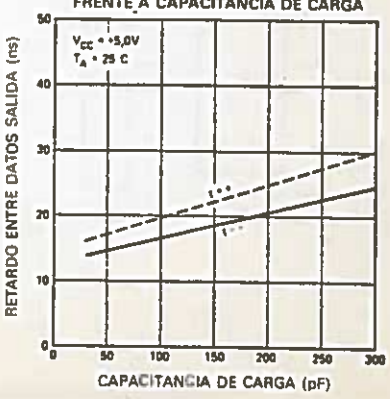
CORRIENTE DE SALIDA FRENTE A TENSION DE SALIDA PARA NIVEL «BAJO»



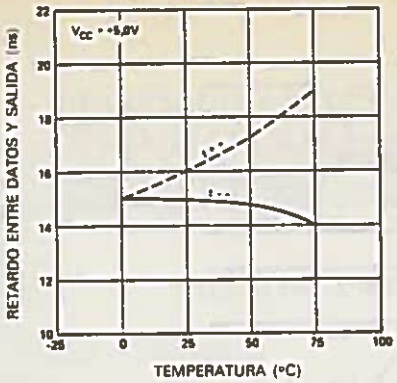
CORRIENTE DE SALIDA FRENTE A TENSION DE SALIDA PARA NIVEL «ALTO»



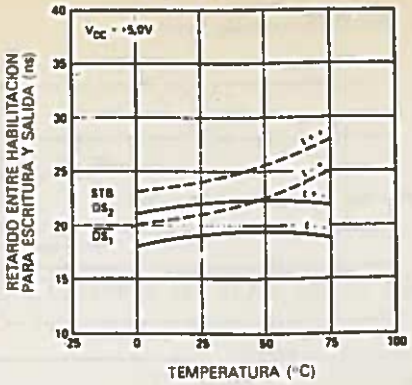
RETARDO ENTRE DATOS Y SALIDA FRENTE A CAPACITANCIA DE CARGA



RETARDO ENTRE DATOS Y SALIDA FRENTE A TEMPERATURA



RETARDO ENTRE HABILITACION PARA ESCRITURA Y SALIDA FRENTE A TEMPERATURA



8212

CARACTERISTICAS EN C.A. $T_A = 0°C$ a $+70°C$, $V_{CC} = +5\ V \pm 5\%$.

Símbolo	Parámetro	Límites			Unidad	Condiciones de prueba
		Mín.	Tip.	Máx.		
t_{PW}	Anchura de impulso	30			ns	
t_{PD}	Retardo entre datos y salida			30	ns	Nota 1
t_{WE}	Retardo entre habilitación para escritura y salida			40	ns	Nota 1
t_{SET}	Tiempo de preparación de los datos	15			ns	
t_H	Tiempo de retención de los datos	20			ns	
t_R	Retardo entre puesta a cero y salida			40	ns	Nota 1
t_S	Retardo entre puesta a uno y salida			30	ns	Nota 1
t_E	Tiempo entre salida habilitación e inhibición			45	ns	Nota 1
t_C	Retardo entre borrado y salida			55	ns	Nota 1

CAPACITANCIA* $F = 1$ MHz, $V_{BIAS} = 2{,}5$ V, $V_{CC} = +5$ V, $T_A = 25°C$.

Símbolo	Prueba	Límites	
		Tip.	Máx.
C_{IN}	Capacitancia de entradas en DS1 y MD	9 pF	12 pF
C_{IN}	Capacitancia de entrada en DS2, CK, ACK, DI_1-DI_8	5 pF	9 pF
C_{OUT}	Capacitancia de salida DO_1-DO_8	8 pF	12 pF

* Este parámetro se muestrea y no se prueba al 100 %.

CARACTERISTICAS DE CONMUTACION

Condiciones de prueba

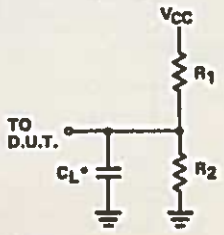
Amplitud del impulso de entrada = 2,5 V
Tiempos de subida y bajada del impulso de entrada = 5 ns.
Las medidas entre 1 y 2 V se hicieron a 1,5 V.
Con una carga de prueba de 15 mA y 30 pF

Nota 1:

Prueba	C_L*	R_1	R_2
t_{PD}, t_{WE}, t_R, t_S, t_C	30 pF	300Ω	600Ω
t_E, autorizado↑	30 pF	10 KΩ	1 KΩ
t_E, autorizado↓	30 pF	300Ω	600Ω
t_E, desautorizado↑	5 pF	300Ω	600Ω
t_E, desautorizado↓	5 pF	10 KΩ	1 KΩ

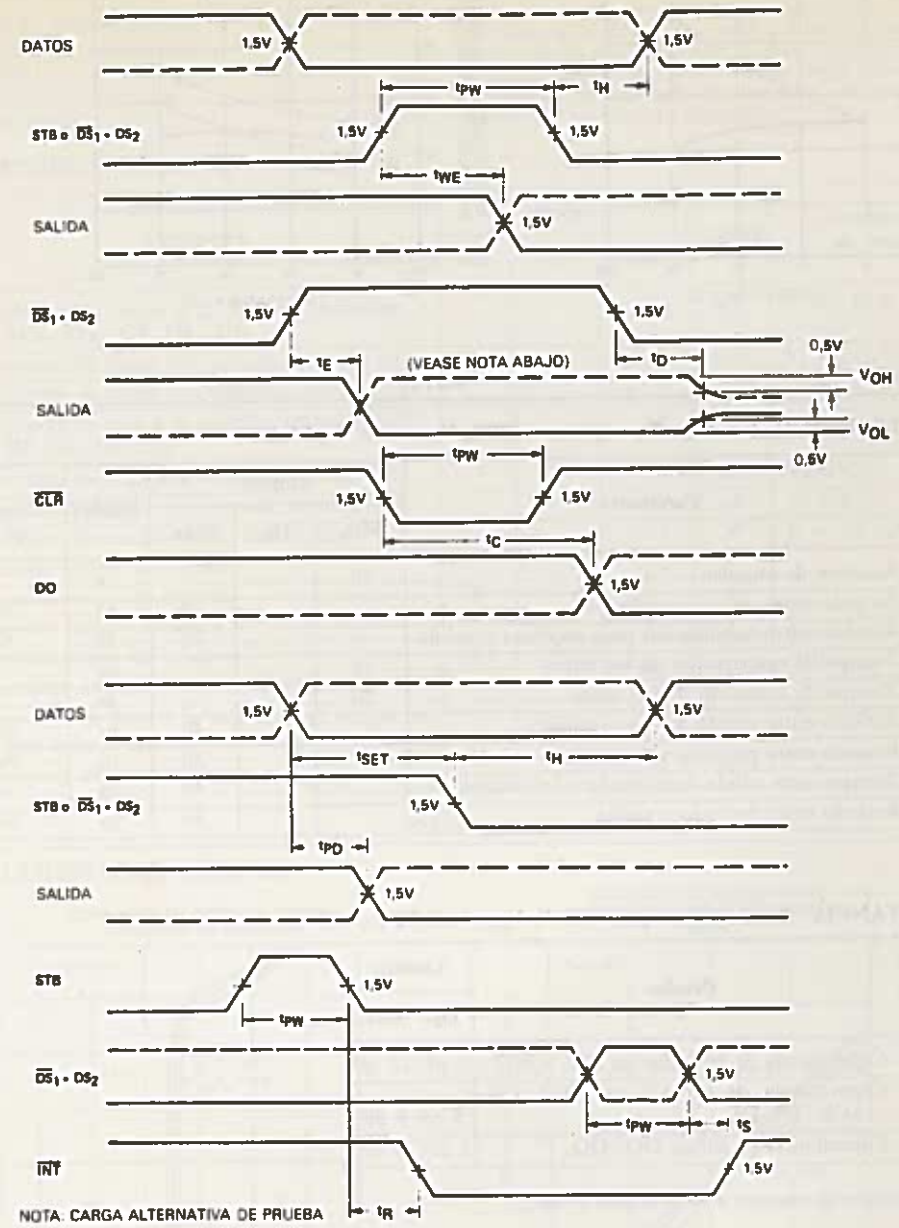
* Incluyendo la capacitancia de la sonda y su soporte.

Carga de prueba
15 mA y 30 pF



*INCLUYENDO LA CAPACITANCIA DE LA SONDA Y SU SOPORTE

DIAGRAMA DE TIEMPOS (cronograma)



Apéndice C6

STANDARD MICROSYSTEMS CORPORATION
35 Marcus Blvd Hauppauge N Y 11787
(516) 273-3100 TWX 510-227-8898
We keep ahead of our competition so you can keep ahead of yours.

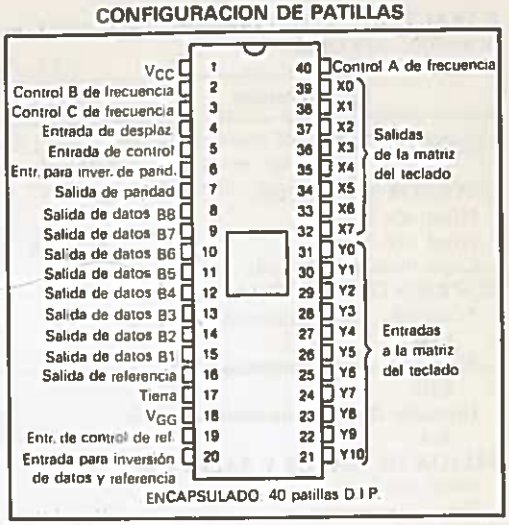
KR2376-XX

MEMORIA ROM DE CODIFICACION DEL TECLADO

CARACTERISTICAS

- **Salidas directamente compatibles con conjuntos lógicos TTL/DTL o MOS.**
- **Dotada de control externo para selección de la polaridad de salida.**
- **Dotada de control externo para selección de paridad par o impar.**
- **Funcionamiento con desplazamiento del cursor mediante dos teclas.**
- **Enclavamiento de cualquier tecla.**
- **Código programable mediante cambio de una sola máscara.**
- **Circuito oscilador incorporado.**
- **Provista de una red de retardo controlada externamente para eliminar el efecto de rebote de los contactos.**
- **Sólo se necesita un circuito integrado para completar el montaje del teclado.**
- **Protección contra la carga estática en todos los terminales de entrada y salida.**
- **Todo el circuito está protegido por una capa de cristal inactiva.**

CONFIGURACION DE PATILLAS



ENCAPSULADO: 40 patillas D I P.

DESCRIPCION GENERAL

La SMC KR2376-XX es una memoria de sólo lectura para 2376 bits con toda la circuitería necesaria para codificar contactos de un teclado unipolar de acción sencilla en forma de palabras utilizables de 9 bits. Los datos y las salidas del impulso de referencia son compatibles directamente con los conjuntos lógicos TTL/DTL o MOS sin necesidad de usar componente especial alguno para interfaz.

La KR2376-XX se fabrica según el procedimiento de canal-P de umbral bajo y contiene transistores tipo 2942 canal-P en modo de enriquecimiento dentro de una patilla monolítica, disponible en encapsulados de 40 patillas en dos líneas DIL.

CONEXION NORMAL DE LA KR2376-XX

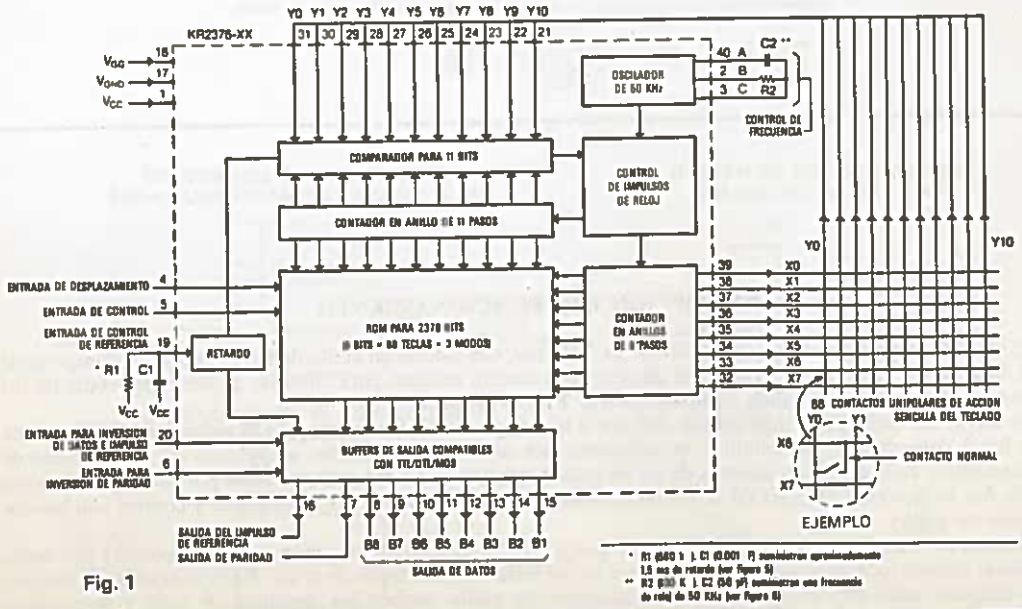


Fig. 1

Impreso con permiso de SMC © 1980.

REGIMENES MAXIMOS GARANTIZADOS*

Margen de temperatura de funcionamiento	de 0°C a +70°C
Margen de temperatura de almacenamiento	de −65°C a +150°C
Tensiones de GND y V_{GG} con respecto a V_{CC}	de −20 V a +0,3 V
Tensiones lógicas de entrada con respecto a V_{CC}	de −20 V a +0,3 V

* Los esfuerzos superiores a los citados con anterioridad pueden ocasionar daño irreparable al dispositivo. La presente es sólo una clasificación de sobrecarga y no implica el funcionamiento del dispositivo bajo estas u otras condiciones por encima de las indicadas en los apartados de funcionamiento de las hojas de características.

CARACTERISTICAS ELECTRICAS (T_A = 0 °C a +70 °C, V_{CC} = 5 V ± 0,5 V, V_{GG} = −12 V ± 1,0 V, salvo que se indique otra cosa).

Características	Mín.	Típ.	Máx.	Unidad	Condiciones
RELOJ	20	50	100	KHz	Véase la nota de la figura 1 (**) para los valores típicos de RC
ENTRADA DE DATOS					
Nivel «0» lógico			+0,8	V	
Nivel «1» lógico	V_{CC}-1,5			V	
Capacitancia de entrada			10	pf	
CORRIENTE DE ENTRADA					
*Control, desplazamiento y Y0 a Y10	10	100	140	μA	V_{IN} = +5,0 V
*Control, desplazamiento y Y0 a Y10	5	30	50	μA	V_{IN} = Tierra
Inversión de datos, inversión de paridad		0,01	1	μA	V_{IN} = −5,0 V a +5,0 V
SALIDA DE DATOS Y SALIDA X					
Nivel «0» lógico			+0,4	V	I_{OL} = 1,6 mA (véase fig. 7)
Nivel «1» lógico	V_{CC}-1,0			V	I_{OH} = 100 μA
DISIPACION DE POTENCIA		140	200	mV	Tensiones nominales de alimentación (véase fig. 8)
CARACTERISTICAS DE CONMUTACION					
Cierre de contacto (tiempo mínimo)	Ver diagrama de tiempos (fig. 2)				
Resistencia a contacto cerrado entre X1 e Y1			300	Ohm	
Resistencia a contacto abierto entre X1 e Y1	1×10^7			Ohm	

* Entradas con resistencia interna a V_{GG}.

DESCRIPCION DEL FUNCIONAMIENTO

La KR2376-XX contiene (véase fig. 1) una ROM de 2376 bits, contadores en anillo de 8 y 11 pasos, un comparador para 11 bits, un circuito oscilador, una red de retardo a control externo para eliminar el efecto de rebote de los contactos y controladores de salida compatibles con TTL/DTL/MOS.

La parte ROM del chip es una memoria de 264 por 9 bits dispuesta en 3 conjuntos de 88 palabras de 9 bits. Cada uno de los 3 conjuntos de 88 palabras se selecciona por medio de los niveles apropiados de las entradas de desplazamiento y control. Las posiciones de las 88 palabras individuales son seleccionadas por los dos contadores en anillo. Así, la dirección de la ROM se forma combinando las entradas de desplazamiento y control con los dos contadores en anillo.

Las salidas al exterior del contador en anillo de 8 pasos y las entradas externas al comparador para 11 bits están cableadas al teclado formando una matriz X-Y con las 88 teclas en los puntos de cruce. En la condición de reserva, cuando ninguna tecla está pulsada, ambos contadores en anillo reciben los impulsos de reloj y seleccionan secuencialmente las direcciones de la ROM; la ausencia de Salida de Referencia indica que las salidas de datos no son válidas en esta ocasión.

Cuando se oprime una tecla, se cierra un camino entre una salida del contador en anillo de 8 pasos (X0 a X7) y una entrada del comparador para 11 bits (Y0 a Y10). Tras un número de ciclos del reloj, ocurrirá una condición según la cual el código del camino seleccionado hacia el comparador casa con el código a la entrada del comparador; que se ve desde el contador en anillo de 11 pasos. Cuando esto ocurre, el comparador envía una señal al control del reloj y a la Salida de Referencia (a través de la red de retardo). El control de reloj detiene los pulsos a los contadores en anillo y las Salidas de Datos se estabilizan con el código de 9 bits (B1-B9) seleccionado, acompañado de una señal de «válido» en la Salida de Referencia. Las Salidas de Datos permanecen estables hasta que se suelta la tecla.

Como característica adicional se suministran dos entradas para control externo de polaridad de las Salidas de Datos. El inversor de paridad (patilla 6) proporciona control de polaridad de la Salida de Paridad (patilla 7), mientras que la Entrada de Inversión de Referencia y Datos suministra control de polaridad sobre las Salidas de Datos B1 a B8 (patillas 8 a 15) y sobre la Salida de Referencia (patilla 16).

SIMBOLOS ESPECIALES

Como el código elegido para cada tecla se define durante la fabricación de la patilla, la codificación puede cambiarse para cubrir cualquier necesidad especial del teclado. Se pueden programar hasta 264 códigos de hasta 8 bits (más uno de paridad) con la ROM KR2376-XX, incluyendo los más populares, tales como los ASCII, EBCDIC, Selectric, etc., así como otros muchos especializados. El código ASCII está disponible como simbolo normalizado. Para símbolos especiales, úsese la figura 9.

DIAGRAMA DE TIEMPOS

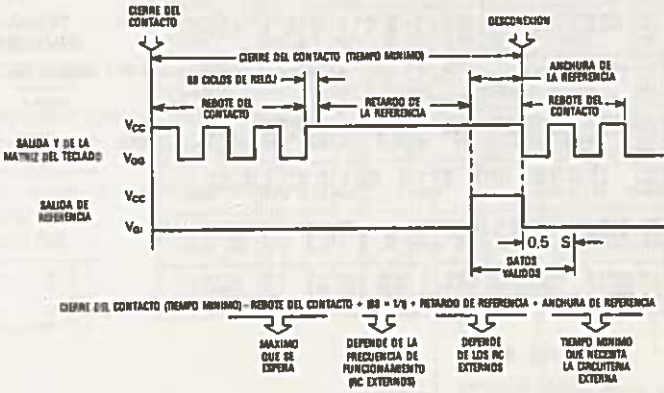
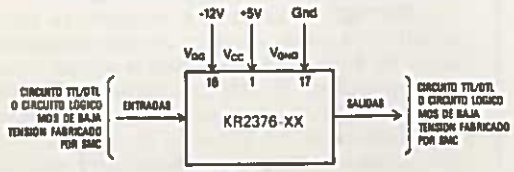
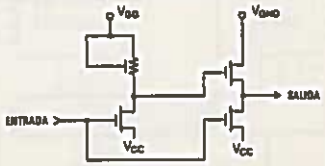


Fig. 2

CONEXIONES DE ALIMENTACION PARA FUNCIONAMIENTO CON TTL/DTL



BUFFER DE SALIDA & ETA DE SALIDA «X» AL TECLADO



CONEXIONES DE ALIMENTACION PARA FUNCIONAMIENTO CON MOS

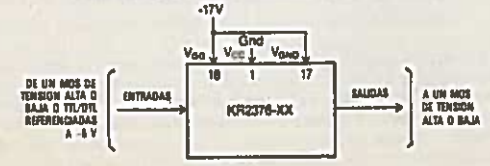


Fig. 3

ETAPA DE ENTRADA «Y» DEL TECLADO

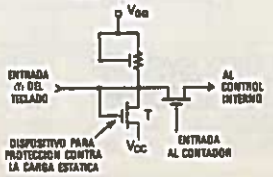


Fig. 4

RETARDO DE LA REFERENCIA FRENTE A C_1

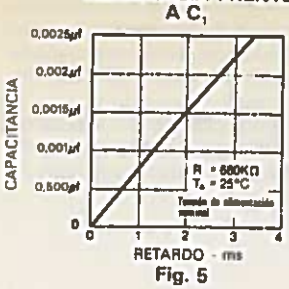


Fig. 5

FRECUENCIA DEL OSCILADOR FRENTE A C_2

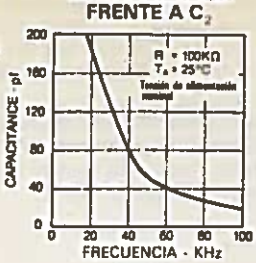


Fig. 6

RESISTENCIA TIPICA DE SALIDA FRENTE A TENSION DE POLARIZACION DE LA PUERTA

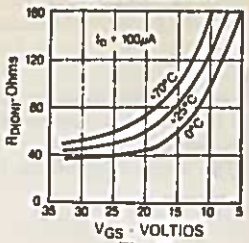


Fig. 7

DISIPACION TIPICA DE POTENCIA FRENTE A TEMPERATURA

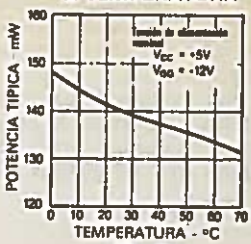


Fig. 8

GRAFICO DE ASIGNACIONES DEL CODIGO
KR2376-ST
ASCII de 8 bits, paridad impar

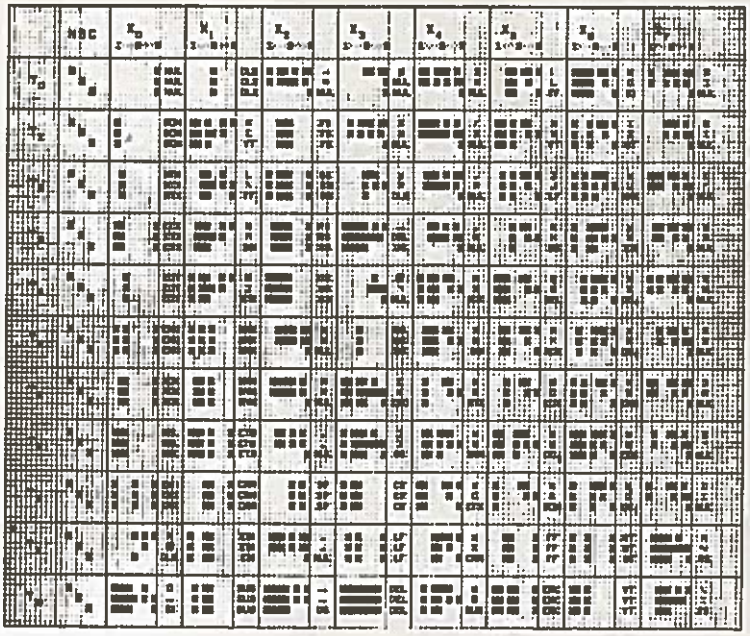


Fig. 9

TABLA DE VERDAD DEL INVERSOR DE DATOS (B1-B8)

ENTRADA DE INVERSION DE REFERENCIA Y DATOS (PATILLA 20)	GRAFICO DE ASIGNACIONES DEL CODIGO	SALIDA DE DATOS (B1-B8)
1	1	0
0	1	1
1	0	1
0	0	0

TABLA DE VERDAD DE INVERSION DE REFERENCIA

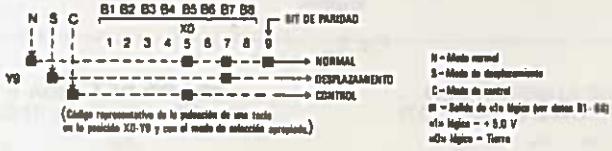
ENTRADA DE INVERSION DE REFERENCIA Y DATOS (PATILLA 20)	REFERENCIA INTERNA	SALIDA DE REFERENCIA (PATILLA 16)
1	1	0
0	0	0
1	0	1
0	1	1

TABLA DE VERDAD DE INVERSION DE PARIDAD

ENTRADA DE INVERSION DE PARIDAD (PATILLA 8)	GRAFICO DE ASIGNACIONES DEL CODIGO	SALIDA DE PARIDAD (PATILLA 7)
1	1	0
0	1	1
1	0	1
0	0	0

SELECCION DE MODO

$\overline{S}$ $\overline{C}$ = N
$\overline{S}$ C = S
S $\overline{C}$ = C
S C = INVALIDO (DATOS ESPUREOS)



STANDARD MICROSYSTEMS CORPORATION

We keep ahead of our competition so you can keep ahead of yours.

Los diagramas de circuitos que utilizan productos SMC se incluyen como medio de ilustración de las aplicaciones típicas de semiconductores; en consecuencia, no es necesario presentar una información completa y suficiente para las necesidades de diseño. La información ha sido comprobada cuidadosamente y se cree que es completamente fiable. Sin embargo, no se asume responsabilidad alguna por las inexactitudes que pudiera haber. Es más, tal información no cede al comprador del dispositivo semiconductor descrito permiso alguno sobre los derechos de patente de SMC u otros. SMC se reserva el derecho a introducir cambios en cualquier momento con el fin de mejorar el diseño y suministrar el mejor producto posible.

Apéndice C7

STANDARD MICROSYSTEMS CORPORATION
35 Marcus Blvd Hauppauge N Y 11787
(516) 273-3100 TWX-510-227-8898
We keep ahead of our competition so you can keep ahead of yours.

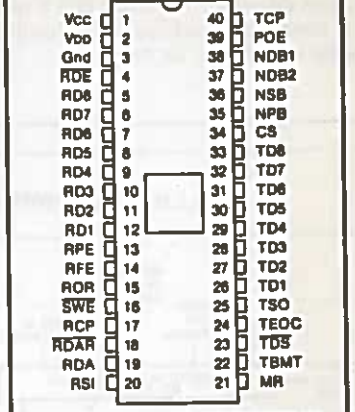
COM 2502
COM 2017
COM 2502/H
COM 2017/H

TRANSMISOR/RECEPTOR ASINCRONO UNIVERSAL (UART)

CARACTERISTICAS

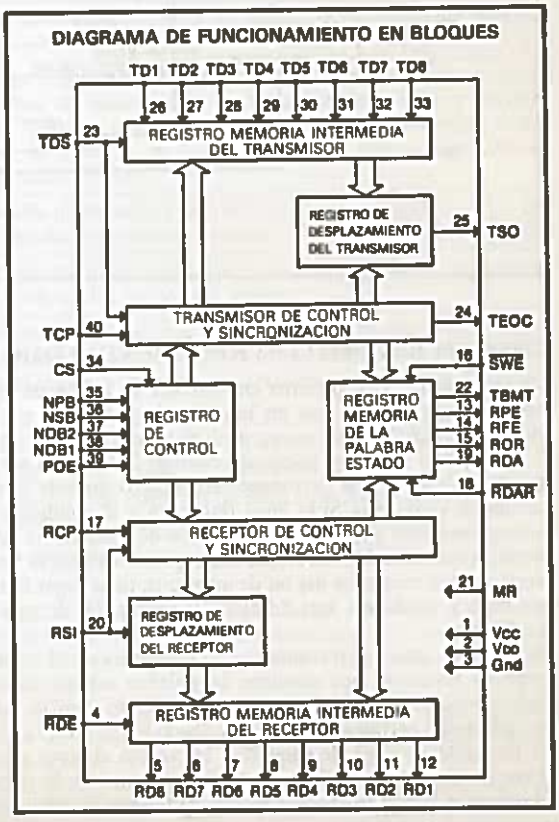
- **Compatibilidad directa con TTL; no se requieren circuitos de interfaz.**
- **Funcionamiento completo en dúplex o semidúplex; puede recibir y transmitir simultáneamente a diferentes velocidades de transmisión de baudios.**
- **Completamente separado por doble partida, lo que elimina la necesidad de sincronización externa exacta.**
- **Verificación del bit de arranque, que disminuye la tasa de errores.**
- **Completamente programable; longitud de la palabra de datos, modo de paridad, número de bits de parada, uno, uno y medio o dos.**
- **Funcionamiento a alta velocidad; 40 kilobaudios, referencias de 200 ns.**
- **Reinicialización principal («maestra»). Pone a cero todas las salidas de estado.**
- **Salidas triestados, diseñadas para una estructura tipo bus.**
- **Bajo consumo; requisitos mínimos de alimentación.**
- **Entrada protegida, lo que elimina problemas de manejo.**
- **Encapsulado DIP cerámico o plástico, para inserción fácil en la tarjeta.**

CONFIGURACION DE PATILLAS



ENCAPSULADO D I P. DE 40 PATILLAS

DIAGRAMA DE FUNCIONAMIENTO EN BLOQUES



DESCRIPCION GENERAL

El transmisor/receptor asíncrono universal es un circuito monolítico MOS/LSI que ejecuta todas las funciones de recepción y transmisión asociadas con las comunicaciones asíncronas de datos. Este circuito se fabrica usando el procedimiento canal-P de óxido nitruro a baja tensión de SMC. El modo dúplex, la velocidad de transmisión en la longitud de la palabra de datos, el modo de paridad y el número de bits de parada son programables independientemente mediante la aplicación de controles del exterior. Puede haber 5, 6, 7 u 8 bits de datos, paridad par/impar o no paridad, y 1 ó 2 bits de parada, o 1,5 bits de parada cuando se utiliza el código de 5 bits de las COM 2017 o COM 2017/H. El UART puede trabajar en cualquiera de los modos dúplex o semidúplex. Estas características programables proporcionan al usuario la posibilidad de interconexión («interfaz») con todos los periféricos asíncronos.

Impreso con permiso de SMC © 1980.

DESCRIPCION DEL FUNCIONAMIENTO COMO TRANSMISOR

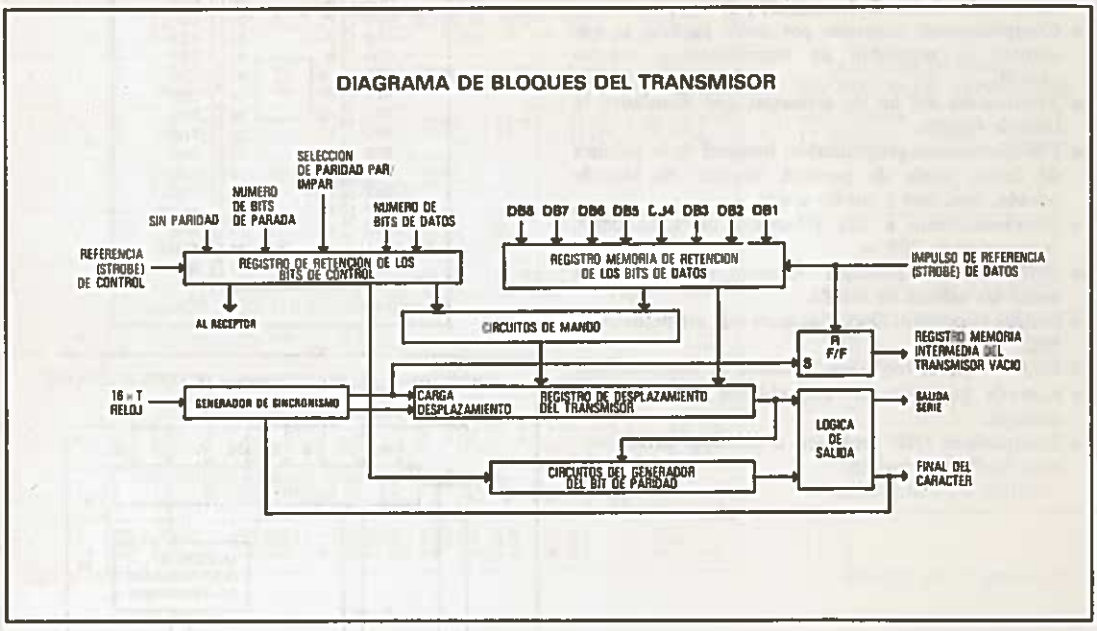
Al encenderlo para ponerlo en marcha, se aplica un tren de impulsos de reloj cuya frecuencia es 16 veces la velocidad de transmisión en baudios deseada y se pulsa la reinicialización (reset) maestra. Bajo estas condiciones TBMT, TEOC y TSO están todas a nivel alto (la línea es activada).

Cuando TBMT y TEOC están a nivel alto, los bits de control pueden estar a uno. Después que esto ocurre los bits de datos pueden estar a uno. Es normal que los bits de control sean introducidos en el transmisor por el impulso de referencia antes que los bits de datos. Sin embargo, y mientras que no se incumplan las especificaciones de anchura mínima del impulso, $\overline{TDS}$ y CS pueden ocurrir simultáneamente. Una vez que el impulso de referencia de datos ($\overline{TDS}$) ha ocurrido, la señal TMBT se va a nivel bajo, indicando que el registro memoria de los bits de datos está lleno y no puede recibir más información.

Si el registro de desplazamiento del transmisor está transmitiendo los datos cargados previamente, la señal TBMT permanece a nivel bajo. Si el registro de desplazamiento del transmisor está vacío, o cuando ha terminado de transmitir el carácter previo, los datos del registro memoria pasan inmediatamente al registro de desplazamiento del transmisor y comienza la transmisión de datos. TSO va a nivel bajo (el bit de arranque), TEOC va a nivel bajo, TBMT va a nivel alto indicando que los datos del registro memoria han sido cargados en el registro de desplazamiento del transmisor y que el primero está disponible para recibir datos nuevos.

Si en este momento se cargan nuevos datos en el registro memoria, TBMT vuelve a nivel bajo y permanece en este estado hasta que se completa la transmisión en curso. Se dispone del tiempo que ocupa la transmisión completa de un carácter para cargar el siguiente carácter sin perder la velocidad de transmisión. Esta es la ventaja de poseer dos registros intermedios.

La transmisión de datos ocurre de una forma ordenada: bit de arranque, bits de datos, bit de paridad (si se la selecciona) y bit(s) de parada. Cuando el último bit de parada ha sido puesto en la línea, TEOC pasa a nivel alto por la duración de un bit. Si TBMT está a nivel bajo, la transmisión comienza inmediatamente. Si TBMT está a nivel alto, el transmisor queda en reposo completo y, si así se desea, se pueden cargar nuevos bits de control antes de la siguiente transmisión de datos.



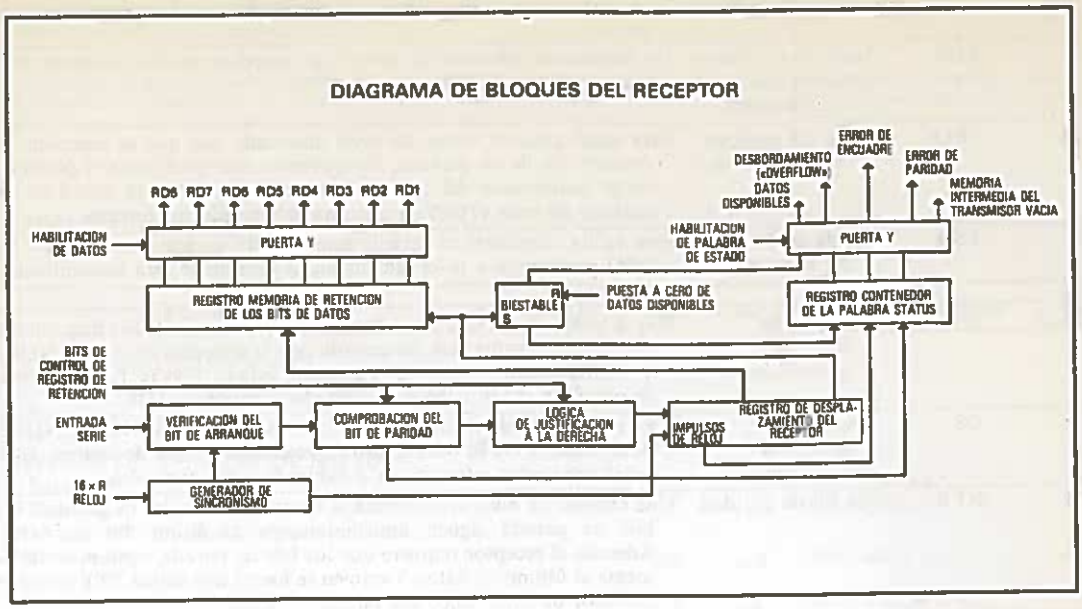
DESCRIPCION DEL FUNCIONAMIENTO COMO RECEPTOR

Al encenderlo para ponerlo en marcha, se aplica un tren de impulsos de reloj cuya frecuencia es 16 veces la velocidad de transmisión en baudios deseada y se pulsa la reinicialización (reset) maestra. La señal de datos disponibles (RDA) está ahora a nivel bajo. Hay sólo un conjunto de bits de control para transmisor y receptor. La recepción de datos comienza cuando la línea de entrada en serie pasa de marca (nivel alto) a espacio (nivel bajo). Si la línea RSI permanece en espacio durante la duración de medio bit, es que se ha verificado un bit de arranque verdadero. Si la línea retornara a la condición de marca (trabajo) antes de que transcurriera el tiempo correspondiente a medio bit, el proceso de verificación del bit de arranque comenzaría de nuevo. Para empezar la verificación del bit de arranque tiene que ocurrir en la línea una transición de marca a espacio. Una vez que se ha verificado la recepción del bit de arranque, tiene lugar la recepción de los datos de forma ordenada: bit de arranque recibido y verificado, bits de datos recibidos, bit de paridad recibido (si se lo ha seleccionado) y bit(s) de parada recibidos.

Si el bit de paridad transmitido no coincide con el recibido, se pone a uno el biestable de error de paridad del registro memoria que contiene la palabra estado, indicando un error de paridad. Sin embargo, si no se ha seleccionado modo de paridad, el biestable de error de paridad se mantiene a cero incondicionalmente, inhibiendo la indicación pertinente. Si no se recibe el bit de parada, debido al encuadre inapropiado del carácter, se pone a uno el biestable de error de encuadre, indicando el error pertinente.

Una vez que se ha recibido un carácter completo, la circuitería lógica interna comprueba el estado de la señal de datos disponibles (RDA). Si en este momento la señal presenta un nivel alto, el receptor entiende que el carácter precedente no ha sido leído y se pone a uno el biestable de desbordamiento («overflow»). La única manera de que el receptor advierta que ha recibido los datos es poner a cero la señal de datos disponibles.

En este momento la salida de RDA se va a nivel alto indicando que todas las salidas pueden ser examinadas. El registro de desplazamiento del receptor está disponible ahora para empezar a recibir el próximo carácter. Dado la existencia de dos registros intermedios, se dispone del tiempo de recepción de un carácter completo para retirar el recibido.

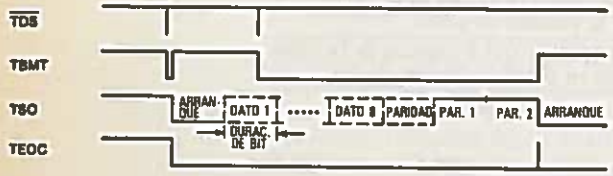


DESCRIPCION DE LAS FUNCIONES DE LA PATILLA

PATILLA NUMERO	SIMBOLO	NOMBRE	FUNCION
1	V_{CC}	Fuente de alimentación	+5 voltios
2	V_{DD}	Fuente de alimentación	−12 voltios
3	GND	Tierra	Tierra
4	$\overline{RDE}$	Hab. para los datos recibidos	Una entrada de nivel bajo autoriza las salidas (RD8-RD1) del registro memoria del receptor.
5-12	RD8-RD1	Dalidas de datos del receptor	Son las 8 salidas triestados habilitadas por $\overline{RDE}$. Las líneas de datos que no se usan, de acuerdo con la selección de NDB1 y NDB2, presentan la salida a nivel bajo y los caracteres recibidos están justificados a la derecha, es decir, que el bit menos significativo aparece en la salida RD1.
13	RPE	Error de paridad en el receptor	La salida triestado (habilitada por $\overline{SWE}$) está a nivel alto si el bit de paridad del carácter recibido no coincide con la paridad seleccionada.
14	RFE	Error de encuadre del receptor	La salida triestado (habilitada por $\overline{SWE}$) está a nivel alto si el carácter recibido no lleva un bit de parada válido
15	ROR	Desbordamiento del receptor	La salida triestado (habilitada por $\overline{SWE}$) está a nivel alto si el carácter recibido anteriormente no fue leído (la salida RDA no se puso a cero) antes de que el carácter en curso sea transferido al registro memoria del receptor.
16	$\overline{SWE}$	Hab. para la palabra de status	Una entrada a nivel bajo habilita las salidas (RPE, RFE, ROR, RDA y TBMT) del registro memoria de la palabra de status.
17	RCP	Reloj del receptor	Esta entrada es un tren de impulsos de reloj cuya frecuencia es 16 veces (16×) la velocidad de transmisión en baudios deseada.
18	$\overline{RDAR}$	Puesta a cero de datos del receptor disponibles	Una entrada de nivel bajo pone a cero la salida RDA.
19	RDA	Datos del receptor disponible	La salida triestado (habilitada por $\overline{SWE}$) está a nivel alto cuando se ha recibido un carácter completo y se lo ha transferido al registro memoria del receptor.
20	RSI	Entrada serie del receptor	Esta entrada acepta la corriente de entrada de bits en serie. Se requiere una transición de nivel alto (marca) a nivel bajo (espacio) para comenzar la recepción de datos.
21	MR	Reinicialización maestra	Esta entrada debe conectarse a nivel alto después de haber encendido el circuito. Ello pone las TSO, TEOC y TBMT a nivel alto, y las RDA, DPE, RFE y ROR a nivel bajo.

PATILLA NUMERO	SIMBOLO	NOMBRE	FUNCION
22	TBMT	Memoria del transmisor vacío	Esta salida triestado (habilitada por SWE) está a nivel alto cuando el registro memoria del transmisor puede ser cargado con información nueva.
23	$\overline{TDS}$	Impulso de referencia de datos en el transmisor	Un impulso de referencia de nivel bajo introduce los bits de datos en el registro memoria del transmisor.
24	TEOC	Final del carácter en el transmisor	Esta señal aparece como un nivel alto cada vez que se completa la transmisión de un carácter. Permanece a este nivel hasta el principio de la transmisión del carácter siguiente, o durante la mitad de un período de reloj (TCP) en el caso de transmisión continua.
25	TSO	Salida en serie del transmisor	Esta salida suministra en serie la totalidad del carácter transmitido. La TSO permanece a nivel alto mientras que no se está transmitiendo datos.
26-33	TD1-TD8	Entradas de datos al transmisor	Hay 8 líneas de entrada de datos (referenciados por TDS) disponibles. Las líneas no utilizadas, de acuerdo con la selección hecha por NDB1 y NDB2, pueden estar en cualquier estado lógico. El bit menos significativo (LSB) debe estar situado siempre en TD1.
34	CS	Impulso de referencia de control	Una entrada de nivel alto llega a los bits de control (NDB1, NDB2, NSB, POE y NPB) del registro de retención de bits de control. Esta señal debe estar referenciada o cableada a un nivel alto.
35	NPB	Sin bit de paridad	Una entrada de nivel alto elimina la transmisión del bit de paridad, los bits de parada siguen inmediatamente al último bit de datos. Además, el receptor requiere que los bits de parada sigan inmediatamente al último de datos. También se fuerza a la salida RPE a tomar un valor de nivel bajo. Ver patilla 39, POE.
36	NSB	Número de bits de parada	Esta entrada selecciona el número de bits de parada. Una entrada de nivel bajo selecciona 1 bit de parada; una de nivel alto selecciona 2 bits. La selección de 2 bits, cuando se ha programado una palabra de 5 bits, origina 1,5 bits de parada en el COM 2017 o en el COM 2017/H.
37-38	NDB2, NDB1	Número de bits de datos por carácter	Estas dos entradas se decodifican internamente para seleccionar 5, 6, 7 u 8 bits de datos por carácter, según la siguiente tabla de verdad: NDB2 NDB1 bits de datos/carácter L L 5 L H 6 H L 7 H H 8
39	POE	Selección de paridad par/impar	El nivel lógico de esta entrada, en conjunción con la de NPB, determina el modo de paridad para receptor y transmisor, de acuerdo con la siguiente tabla de verdad: * NPB POE MODO L L Paridad impar L H Paridad impar H X Sin paridad no afecta
40	TCP	Reloj de transmisor	Esta entrada es un tren de impulsos de reloj cuya frecuencia es 16 veces (16×) la velocidad de transmisión en baudios que se desea.

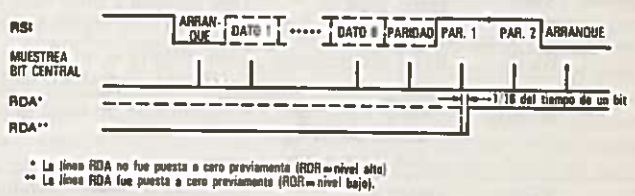
SINCRONIZACION DEL TRANSMISOR - 8 BITS, PARIDAD, 2 BITS DE PARADA



ARRANQUE DEL TRANSMISOR

A la iniciación de la transmisión de datos, o cuando las líneas no se utilizan al 100%, en la transmisión, el bit de arranque se coloca en la línea TSO a la transición de nivel alto a nivel bajo del reloj TCP, inmediatamente después del flanco posterior de TDS.

RECEPCION DEL TRANSMISOR-8 BITS, PARIDAD, 2 BITS DE PARADA



* La línea RDA no fue puesta a cero previamente (ROR = nivel alto)
** La línea RDA fue puesta a cero previamente (ROR = nivel bajo).

DETECCION/VERIFICACION DEL BIT DE ARRANQUE

Si la línea RSI permanece en espacio durante el tiempo de 1/2 bit, se verifica un bit de arranque auténtico. Si la línea volviera a la condición de marca (trabajo) antes de que transcurriera el tiempo de 1/2 bit, el proceso de verificación del bit de arranque comenzaría de nuevo.

REGIMENES MAXIMOS GARANTIZADOS*

Margen de temperatura de funcionamiento	de 0 °C a +70 °C
Margen de temperatura de almacenamiento	de −55 °C a +150 °C
Temperatura del estaño (soldadura en 10 seg)	+325 °C
Tensión positiva en cualquier patilla, V_{CC}	+0,3 V
Tensión negativa en cualquier patilla, V_{CC}	−25 V

* Los esfuerzos superiores a los citados con anterioridad pueden ocasionar daño irreparable al dispositivo. La presente es sólo una clasificación de sobrecarga y no implica el funcionamiento del dispositivo bajo estas u otras condiciones por encima de las indicadas en los apartados de funcionamiento de las hojas de características.

CARACTERISTICAS ELECTRICAS (T_A = 0 °C a 70 °C, V_{CC} = +5 V ± 5 %, V_{DD} = −12 V ± 5 %, salvo que se indique otra cosa).

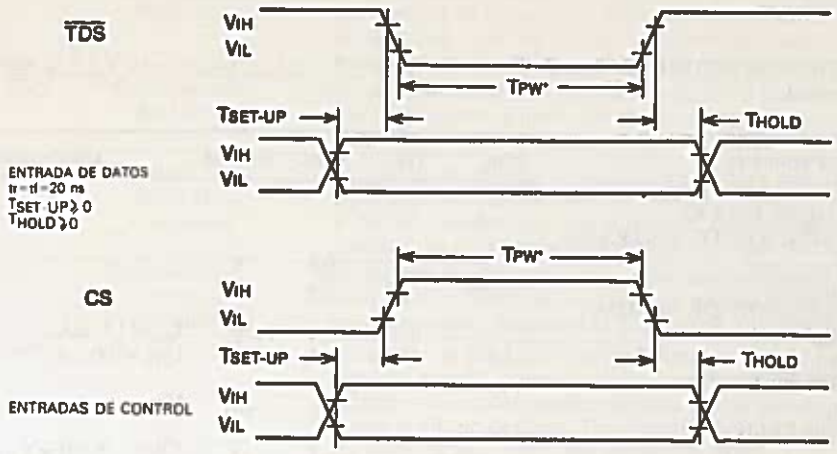
Parámetro	Mín.	Típ.	Máx.	Unidad	Condiciones
CARACTERISTICAS EN C.C.					
NIVELES DE TENSION DE ENTRADA					
Nivel bajo, V_{IL}	V_{DD}		0,8	V	
Nivel alto, V_{IH}	V_{CC}-1,5		V_{CC}	V	
NIVELES DE TENSION DE SALIDA					
Nivel bajo, V_{OL}		0,2	0,4	V	I_{OL} = 1,6 mA
Nivel alto, V_{OH}	2,4	4,0		V	I_{OH} = 100 μA
CORRIENTE DE ENTRADA					
Nivel bajo, I_{IL}			1,6	mA	Ver nota 4
CORRIENTE DE SALIDA					
Fuga, I_{LO}			−1	μA	$\overline{SWE} = \overline{RDE} = V_{IH}$, $0 \leq V_{OUT} \leq +5$ V
Cortocircuito, I_{OS}**			10	mA	V_{OUT} = 0 V
CAPACITANCIA DE ENTRADA					
Todas las entradas, C_{IN}		5	10	pf	$V_{IN} = V_{CC}$, f = 1 MHz
CAPACITANCIA DE SALIDA					
Todas las salidas, C_{OUT}		10	20	pf	$\overline{SWE} = \overline{RDE} = V_{IH}$, f = 1 MHz
CORRIENTE DE ALIMENTACION					
I_{CC}			28	mA	Todas las salidas = V_{OH}, todas las entradas = V_{CC}
I_{DD}			28	mA	
CARACTERISTICAS EN C.A.					T_A = +25 °C
FRECUENCIA DEL RELOJ					
(COM 2502, COM 2017)	DC		400	KHz	RCP, TCP
(COM 2502/H, COM 2017/H)	DC		640	KHz	RCP, TCP
ANCHURA DEL IMPULSO					
Reloj	1			μs	RCP, TCP
Puesta a cero maestra	500			ns	MR
Referencia de control	200			ns	$\overline{CS}$
Referencia de datos del transmisor	200			ns	$\overline{TDS}$
Puesta a cero de datos disponibles en el receptor	200			ns	$\overline{RDAR}$
TIEMPO DE PREPARACION					
Bits de datos	≥0			ns	TD1-TD8
Bits de control	≥0			ns	NPB, NSB, NDB2, NDB1, POE
TIEMPO DE RETENCION A LA ENTRADA					
Bits de datos	≥0			ns	TD1-TD8
Bits de control	≥0			ns	NPB, NSB, NDB2, NDB1, POE
RETARDO ENTRE REFERENCIA Y SALIDA					Carga = 20 pf + 1 TTL input
Habilitación para recibir datos			350	ns	$\overline{RDE}$: T_{PD1}, T_{PD0}
Habilitación de la palabra de estado			350	ns	$\overline{SWE}$: T_{PD1}, T_{PD0}
RETARDO PARA LA DESACTIVACION DE LA SALIDA			350	ns	$\overline{RDE}$, $\overline{SWE}$

** No debe cortocircuitarse más que una salida cada vez.

NOTAS:

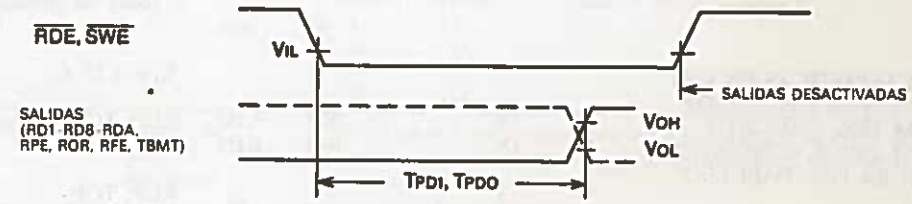
1. Si el transmisor está inactivo (TEOC y TBMT están a nivel alto) el bit de arranque aparecerá en la línea TSO dentro de un periodo del reloj (TCP) a partir del flanco posterior de TDS.
2. El bit de arranque (transición de señal a espacio) será siempre detectado dentro de un periodo de reloj a partir de RCP, garantizando un deslizamiento máximo del bit de arranque de 1/16 del tiempo de un bit.
3. La salida triestado dispone de 3 estados: 1) baja impedancia a V_{CC}, 2) baja impedancia a GND y 3) alta impedancia OFF ≅ 10 MΩ. El estado OFF está controlado por las entradas $\overline{SWE}$ y $\overline{RDE}$.
4. Bajo las condiciones del estado estacionario no fluye corriente por los interfaces TTL o MOS (COM 2502 o COM 2502/H).

DIAGRAMA DE TIEMPOS DE DATOS/CRONOGRAMA

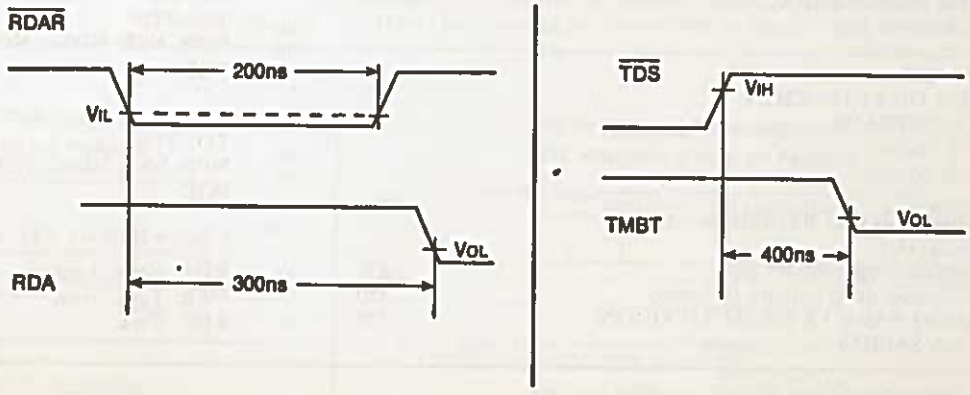


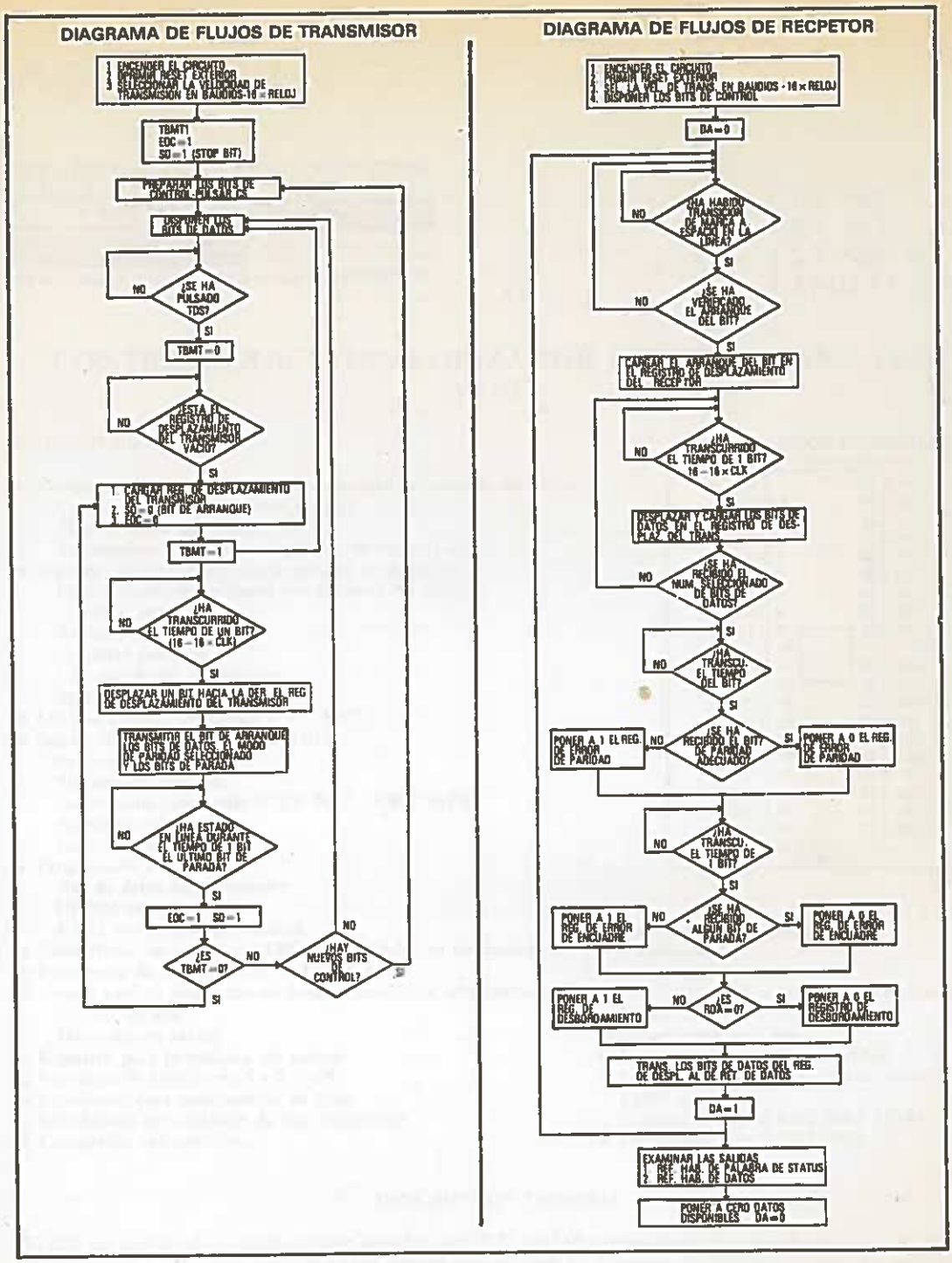
* La información de entrada (datos/control) sólo necesita ser válida durante la última Tpw. tiempo mínimo de las referencias de entrada (TDS, CS)

DIAGRAMA DE TIEMPOS PARA LA SALIDA (CRONOGRAMA)



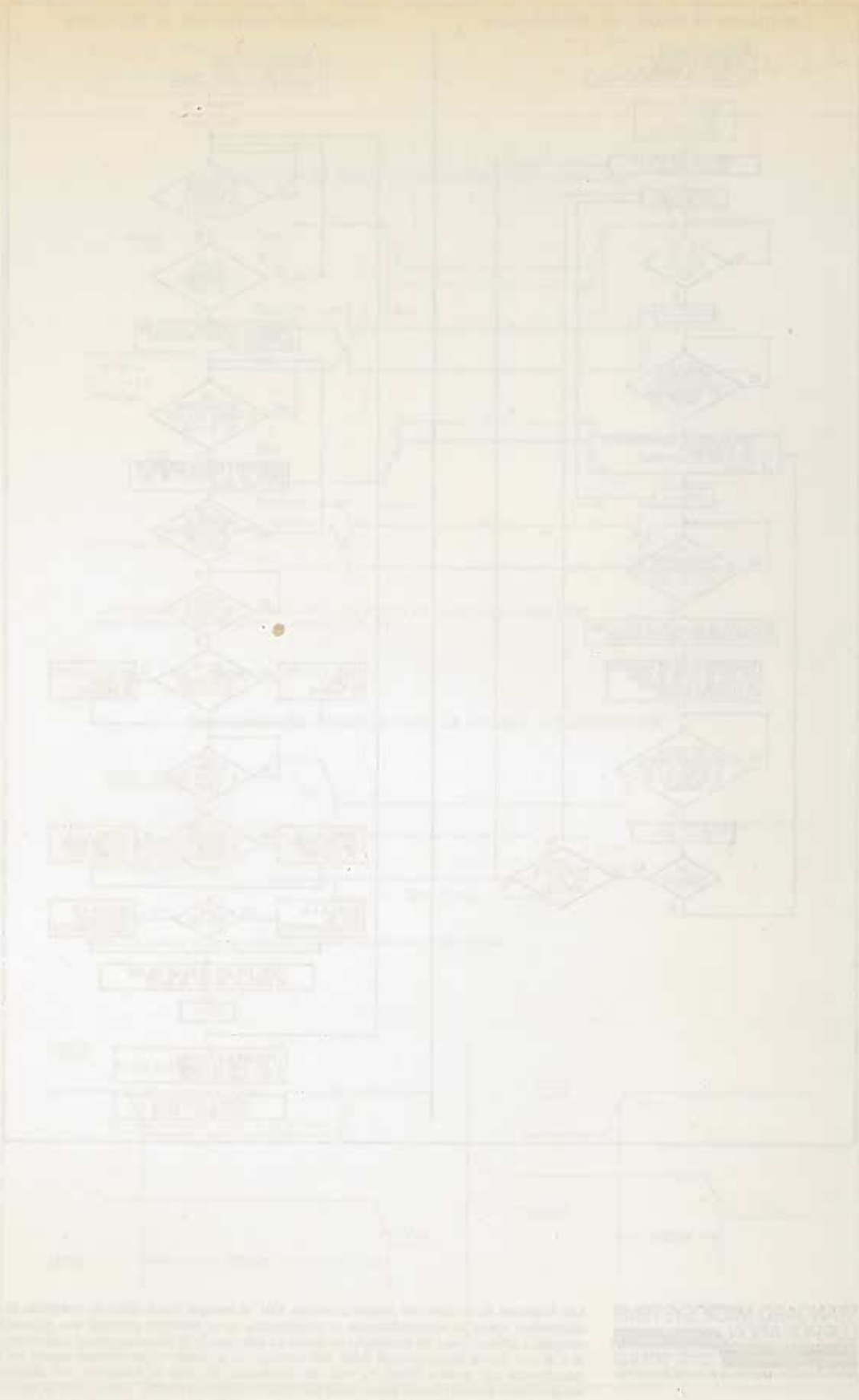
NOTA: Los dibujos de formas de onda no están a escala para mejor claridad.





STANDARD MICROSYSTEMS CORPORATION

Los diagramas de circuitos que utilizan productos SMC se incluyen como medio de ilustración de las aplicaciones típicas de semiconductores; en consecuencia, no es necesario presentar una información completa y suficiente para las necesidades de diseño. La información ha sido comprobada cuidadosamente y se cree que es completamente fiable. Sin embargo, no se asume responsabilidad alguna por las inexactitudes que pudiera haber. Es más, tal información no cede al comprador del dispositivo semiconductor descrito permiso alguno sobre los derechos de patente de SMC u otros. SMC se reserva el derecho a introducir cambios en cualquier momento con el fin de mejorar el diseño y suministrar el mejor producto posible.



Apéndice C8

STANDARD MICROSYSTEMS CORPORATION
35 Marcus Blvd Hauppauge N Y 11787
(516) 273 3100 TWX 510-227 8898
We keep ahead of our competition so you can keep ahead of yours.

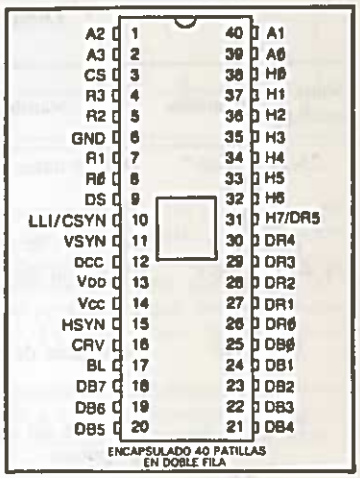
CRT 5027
CRT 5037
CRT 5057*
FAMILIA μPC

CONTROLADOR Y TEMPORIZADOR DE VIDEO PARA TRC VTAC

CARACTERISTICAS

- **Formato completamente programable para la presentación visual:**
  - **Caracteres por fila de datos (1-200).**
  - **Filas de datos por cuadro (1-64).**
  - **Exploraciones de la trama por fila de datos (1-16).**
- **Formato de sincronismo para monitor programable:**
  - **Exploraciones de la trama por cuadro (256-1023).**
  - **«Umbral anterior».**
  - **Anchura de sincronismo.**
  - **«Umbral posterior».**
  - **Entrelazado/no entrelazado.**
  - **Supresión de línea.**
- **Entrada de línea enclavada (CRT 5057).**
- **Salidas directas para monitor TRC:**
  - **Sincronismo horizontal.**
  - **Sincronismo vertical.**
  - **Sincronismo compuesto (CRT 5027 y CRT 5037).**
  - **Supresión del haz.**
  - **Coincidencia de cursor.**
- **Programado a través de:**
  - **Bus de datos del procesador.**
  - **PROM externa.**
  - **ROM con opción de máscara.**
- **Compatible con monitores TRC normalizados/no normalizados.**
- **Frecuencia de regeneración: 60 Hz, 50 Hz,...**
- **Avance vertical progresivo de líneas («scrolling», arrollamiento):**
  - **De una en una.**
  - **De varias en varias.**
- **Registros para la posición del cursor.**
- **Formatos de caracteres: 5 × 7, 7 × 9.**
- **Posicionamiento programable de filas.**
- **Entrelazado por corriente de haz equilibrada.**
- **Compatible con gráficos.**

CONFIGURACION DE PATILLAS



ENCAPSULADO 40 PATILLAS EN DOBLE FILA

- **Aplicaciones de pantalla partida:**
  - **Horizontal.**
  - **Vertical.**
- **Funcionamiento entrelazado/no entrelazado.**
- **Compatible con TTL.**
- **Encaminado para bus.**
- **Funcionamiento a alta velocidad.**
- **Tecnología de puesta de silicio canal-N COPLAMOS.**
- **Compatible con el CRT 8002 VDAC™.**
- **Compatible con el CRT 7004.**

DESCRIPCION GENERAL

El chip del controlador y temporizador de vídeo para TRC (VTAC) es un dispositivo programable por el usuario de 40 patillas y MOS/LSI canal-N COPLAMOS que contiene las funciones lógicas para generar las señales de tiempo para la presentación y el formato de datos de vídeo entrelazados, o no entralazados, sobre un monitor de TRC normalizado o no normalizado.

Con la excepción del contador de puntos, que puede ser alimentado con videofrecuencia de más de 25 MHz y por ello no resulta recomendable construirlo con MOS, todo el formato del cuadro, tal como los sincronismos horizontal, vertical y compuesto, caracteres por fila de datos, filas de datos por cuadro y exploraciones de la traza por cuadro y por fila son completamente programables por el usuario. El contador de filas ha sido diseñado para facilitar el avance progresivo de filas.

La programación se efectúa cargando siete registros de control de 8 bits directamente fuera del bus bidireccional de 8 bits para datos. Cuatro líneas de direccionamiento de registros y una línea de selección de patilla suministran compatibilidad total con microprocesadores para la preparación controlada del programa. El dispositivo puede ser autocargado mediante PROM exterior conectada al bus de datos, como se describe en el apartado de FUNCIONAMIENTO. El formato puede ser también programado por una sola opción de máscara.

Además de los siete registros de control, se han dispuesto otros dos registros adicionales para almacenar el carácter del cursor y las direcciones de las filas de datos para la generación de la señal de vídeo del cursor. El contenido de estos dos registros puede también leerse sobre el bus para poner el programa al dia.

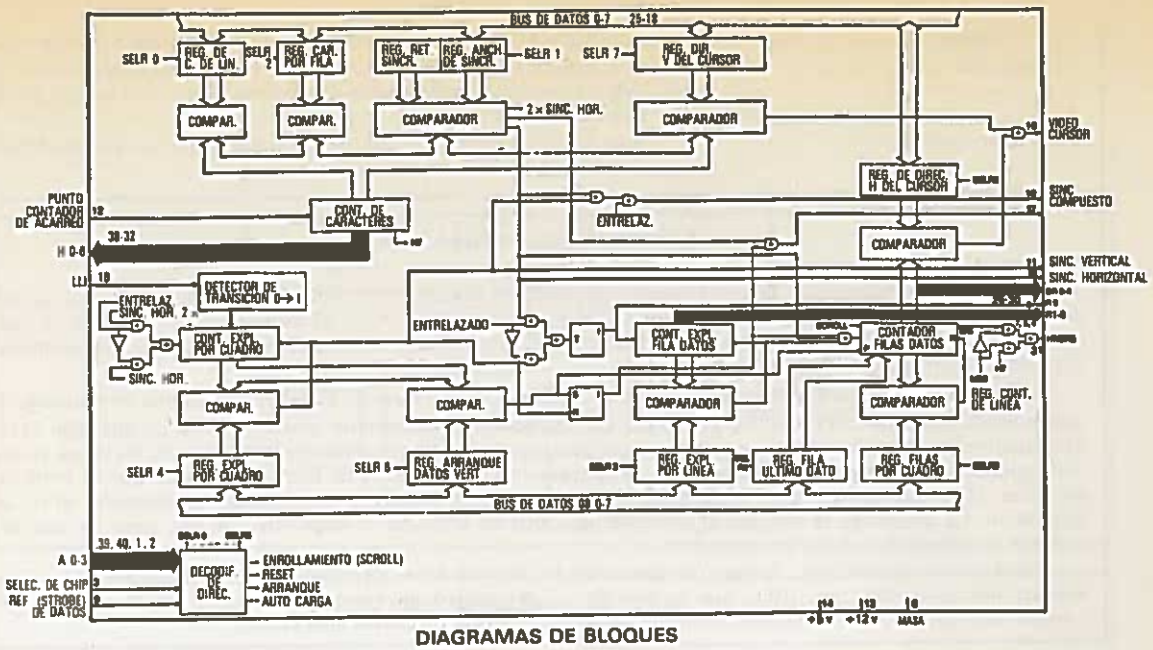
* LANZAMIENTO PROXIMO

Impreso con permiso de SMC © 1980.

Hay disponibles tres versiones del VTAC. La CRT 5027 proporciona funcionamiento no entrelazado con un número par o impar de líneas exploradas por fila de datos, o funcionamiento entrelazado con un número par de exploraciones por fila de datos. El CRT 5037 puede ser programado para un número par o impar de líneas de exploración por fila, en ambos modos entrelazado y no entrelazado. La programación del CRT 5037 para un número impar de líneas de exploración por fila elimina la distorsión de caracteres introducida por la desigual corriente de haz asociada normalmente con el entrelazado del campo par con el entrelazado del campo impar con el campo par de las presentaciones visuales alfanuméricas.
El CRT 5057 proporciona la posibilidad de fijar la frecuencia de regeneración vertical del TRC, cuando está controlada por el impulso de sincronismo vertical del VTAC, a los 50 ó 60 Hz de la frecuencia de la línea, eliminando así el llamado fenómeno «ondulación». Esto está especialmente bien adecuado a los requisitos de los sistemas europeos. La forma de onda de la frecuencia de la línea, procesada para conformar los niveles lógicos especificados por el VTAC, se aplica a la entrada de enclavamiento de línea. El VTAC anulará la generación de sincronismo vertical que detecte en esta entrada una transición de cero a uno. Se comienza entonces con el pulso de sincronismo vertical dentro de una línea de exploración después de que esta transición se eleve por encima del umbral lógico del VTAC.
Para suministrar la patilla necesaria para la entrada de enclavamiento de líneas, la salida del sincronismo compuesto no se suministra en el CRT 5057.

Descripción de las funciones de las patillas

Núm. de patilla	Símbolo	Nombre	Entrada/ Salida	Función
25-18	DBØ-7	Bus de datos	E/S	Bus de datos. Bus de entrada para palabras de control del microprocesador o PROM. Bus bidireccional para la dirección del cursor.
3	CS	Selección de chip	E	Señala el chip que está siendo seleccionado.
39, 40, 1, 2	AØ-3	Dirección del registro	E	Bits de dirección del registro para selección de uno de los siete registros de control o uno de los registros de dirección del cursor.
9	$\overline{DS}$	Referencia de datos	E	Referencia DBØ7 en el registro apropiado, o saca la dirección del carácter del cursor o la dirección de línea del cursor al bus de datos.
12	DCC	Arrastre del contador de puntos	E	Arrastre del contador de puntos externo que establece la frecuencia de reloj de los caracteres básicos. Reloj de caracteres.
38-32	HØ-6	Salidas del contador de caracteres	S	Salidas del contador de caracteres.
7, 5, 4	R1-3	Salidas del contador de barridos	S	Los 3 bits más significativos del contador de barrido; entradas de selección de fila al generador de caracteres.
31	H7/DR5	H7/DR5	S	La definición de patillas es programable por el usuario. La salida es el bit más significativo (NSB) del contador de caracteres si la cuenta es de líneas horizontales (REG/ Ø) es $\geqslant$128; en otro caso, la salida es el bit más significativo del contador de filas de datos.
8	RØ	Bit menos significativo (LSB) del contador de barrido	S	Bit menos significativo del contador de barrido. En el modo de entrelazado, con un número par de exploraciones por fila, RØ cambiará de estado al ritmo de los campos; para un número impar de exploraciones por fila de datos en el modo de entrelazado, RØ cambiará al ritmo de la fila de datos.
26-30	DRØ-4	Salidas del contador de filas de datos	S	Salidas del contador de filas de datos.
17	BL	Borrado	S	Define la parte no activa de las exploraciones vertical y horizontal.
15	HSYN	Sincronismo horizontal	S	Empieza el retroceso horizontal.
11	VSYN	Sincronismo vertical	S	Empieza el retroceso vertical.
10	CSYN/ LLI	Salida de sincronismo compuesto/entrada de enclavamiento de líneas	S/E	El sincronismo compuesto se suministra en los CRT 5027 y CRT 5037. Esta salida está activada en el modo de no entrelazado solamente. Proporciona una forma de onda de sincronismo compuesto exacta a la del RS-170. En el CRT 5057, esta patilla es la entrada de enclavamiento de línea. La forma de onda de la frecuencia de línea, procesada para conformar los niveles lógicos específicos del VTAC, se aplica a esta patilla.
16	CRV	Vídeo cursor	S	Define la posición del cursor en el campo de datos.
14	V_{CC}	Fuente de alimentación	PS	Tensión de alimentación de +5 V.
13	V_{CC}	Fuente de alimentación	PS	Tensión de alimentación de +12 V.



DIAGRAMAS DE BLOQUES

Funcionamiento

La filosofía empleada en el diseño fue permitir que el dispositivo hiciera interfaz efectiva con un sistema lógico basado en microprocesador o de lógica cableada. El dispositivo es programable por el usuario de dos maneras: mediante bus de datos del microprocesador como parte de la rutina de iniciación del sistema, o durante el encendido, mediante una PROM conectada al bus de datos y direccionada directamente por las salidas de selección de fila del chip (ver figura 4). Se necesitan siete palabras de 8 bits para programar completamente el dispositivo. La asignación de bits para estas palabras se muestra en la tabla 1. La información contenida en estas siete palabras consiste en lo siguiente:

Formato horizontal:

Caracteres/fila — Un código de 3 bits que suministra 8 longitudes de caracteres programables por máscara de 20 a 132. Un dispositivo normalizado quedará enmascarado por las siguientes longitudes de carácter: 20, 32, 40, 64, 72, 80, 96 y 132.

Retardo del sincronismo horizontal — Los 3 bits asignados suministran hasta 8 tiempos de carácter para formar el «umbral anterior».

Anchura del sincronismo horizontal — Los 4 bits asignados suministran hasta 15 tiempos de carácter para formar la anchura del sincronismo horizontal.

Cuenta de líneas horizontales — Los 8 bits asignados suministran hasta 256 tiempos de carácter para formar la anchura del sincronismo horizontal.

Bits de sesgo — Un código de 2 bits que suministra de 0 a 2 caracteres de sesgo (skew) entre el contador de dirección horizontal y las señales de borrado y sincronismo (horizontal, vertical o compuesto) para permitir el ajuste en el tiempo de los datos de vídeo antes de generar la señal compuesta de vídeo. La señal de Vídeo Cursor está también sesgada como función de este código.

Formato vertical:

Entrelazado/no entrelazado — Este bit influye en la presentación de los datos con formato de campo par/impar en los sistemas entrelazados. Modifica los contadores de sincronización vertical como se describe más adelante. Un «1» lógico establece el modo de entrelazado.

Exploraciones/cuadro — Son 8 bits los asignados, definidos de acuerdo con las siguientes ecuaciones: tomemos X = valor de los 8 bits asignados.
1) En el modo entrelazado/exploraciones/cuadro = 2X + 513. Por tanto, para 525 exploraciones, programar X = 6 (00000110). El sincronismo vertical ocurrirá exactamente cada 262,5 exploraciones, produciendo así dos campos entrelazados.
Alcance = 513 a 1023 exploraciones/cuadro, cuentas impares únicamente.
2) En el modo no entrelazado/exploraciones/cuadro = 2X + 256. Por tanto, para 262 exploraciones, programar X = 2 (00000011).
Alcance = 256 a 766 exploraciones/cuadro, cuentas pares únicamente. En cualquier modo, la anchura del sincronismo vertical está fijada a 3 exploraciones horizontales (=3H).

Arranque de datos verticales — Son 8 bits que definen el número de exploraciones de la traza desde el flanco anterior del impulso de sincronismo vertical hasta el inicio de los datos representados. En esta exploración de la traza el contador de filas de datos contiene la dirección de la fila del borde superior de la página.

Filas/cuadro	Son 6 bits los asignados que suministran hasta 64 filas por cuadro.
Ultima fila	Son 6 bits para permitir el avance o retroceso vertical progresivo de líneas mediante una definición previa a su carga de la cuenta de la última fila representada.
Exploraciones/fila	Son 4 bits asignados que suministran hasta 16 líneas de exploración por fila de datos.

Características adicionales

Iniciación del dispositivo:

Bajo control de microprocesador. El dispositivo puede ser puesto en las condiciones iniciales bajo control del sistema o de programa mediante preselección de la dirección 1Ø1Ø en A3-Ø. El dispositivo permanecerá en este estado, posicionado sobre la parte superior de la página del campo par, hasta que se ejecute una orden de arranque al presentar la dirección 1Ø1Ø en A3-Ø.

Mediante «autocarga». En un sistema no computarizado, la secuencia de autocarga se efectúa presentando y reteniendo la dirección 1111 en A3-Ø y se inicia a la recepción del impulso de referencia (DS). La dirección 1111 debe mantenerse lo suficiente para asegurar que los siete registros se han cargado (en la mayoría de los casos, en un milisegundo). La secuencia de sincronización comenzará en la exploración de líneas después de que se retire la dirección 1111. En los sistemas computarizados, la autocarga comienza presentando la dirección Ø111 al dispositivo. La autocarga se termina al presentar la orden de arranque al dispositivo, lo que pone en marcha también la cadena de señales de tiempo.

Enrollamiento («scrolling»). Además de almacenar los últimos datos representados en el registro 6, una orden enrollar («scroll») (dirección 1Ø11) que se presente al dispositivo incrementará la cuenta de la primera fila representada para facilitar el arrollamiento de líneas hacia arriba en ciertas aplicaciones.

Diagrama de programación de los registros de control

Cuenta de líneas horizontales: Total de caracteres/línea = N + 1, N = 0 a 255 (DB0 = LSB*)

Caracteres fila de datos:

DB2	DB1	DB0			
0	0	0	=	20	caracteres activos/fila
0	0	1	=	32	
0	1	0	=	40	
0	1	1	=	64	
1	0	0	=	72	
1	0	1	=	80	
1	1	0	=	96	
1	1	1	=	132	

Retardo de sincronismo horizontal: = N, de 1 a 7 tiempos de carácter (DB0 = LSB) (N = 0 prohibido)

Anchura de sincronismo horizontal: = N, de 1 a 15 tiempos de carácter (DB3 = LSB) (N = 0 prohibido)

Bits de sesgo (skew):

DB7	DB8	Retardo sincro/borrado (Tiempos de carácter)	Retardo cursor
0	0	0	0
1	0	1	0
0	1	2	1
1	1	2	2

Barridos/cuadro: 8 bits asignados, definidos de acuerdo con las ecuaciones siguientes:
Sea X = valor de los 8 bits asignados (DB0 = LSB).
1) Modo entrelazado; exploraciones/cuadro = 2X + 513.
Por tanto, para 525 exploraciones, programar X = 6 (00000110). El sincronismo vertical ocurrirá exactamente cada 262,5 exploraciones, produciendo así dos campos entrelazados.
Margen = 513 a 1023 exploraciones/cuadro, cuentas impares solamente.
2) Modo no entrelazado; exploraciones/cuadro = 2X + 256. Por tanto, para 262 exploraciones, programar X = 3 (00000011).
Margen = 256 a 766 exploraciones/cuadro, cuentas pares solamente.
En ambos modos, la anchura del sincronismo vertical está fijada a 3 exploraciones horizontales (= 3 H).

Arranque de datos verticales: N = número de retardo de líneas de traza después flanco anterior del sincronismo vertical de la posición de arranque vertical (DB0 = LSB).

Filas/cuadro: Número de filas = N + 1, N = 0 a 63 (DB0 = LSB).

Ultima fila: N = Dirección de la última fila representada.
N = 0 a 63, pie para 24 filas, programar N = 23 (DB0 = LSB).

Modo: Registro 1, DB7 = 1 establece entrelazado.

Exploraciones/fila:

Modo entrelazado

CRT 5027: exploraciones/fila = N + 1, donde N = número de filas programadas. N = 0 a 15. Las exploraciones por fila tienen que ser cuentas pares solamente.
CRT 5037, CRT 5057: exploraciones/fila = N + 2. N = 0 a 14 cuentas pares o impares.

Modo no entrelazado

CRT 5027, CRT 5037, CRT 5057: exploraciones por fila = N + 1, cuenta par o impar N = 0 a 15.

* LSB = Bit menos significativo.

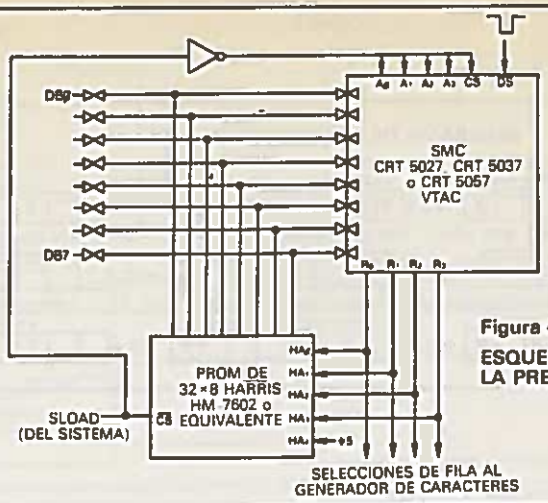


Figura 4
ESQUEMA DE AUTOCARGA PARA LA PREPARACION DEL VTAC

Códigos para selecciones de registros/órdenes

A3	A2	A1	Aø	Selección/Orden	Descripción
0	0	0	0	Cargar Registro de Control ø	Ver tabla 1
0	0	0	1	Cargar Registro de Control 1	
0	0	1	0	Cargar Registro de Control 2	
0	0	1	1	Cargar Registro de Control 3	
0	1	0	0	Cargar Registro de Control 4	
0	1	0	1	Cargar Registro de Control 5	
0	1	1	0	Cargar Registro de Control 6	
0	1	1	1	Autocarga comenzada por el microprocesador	Orden del procesador instruyendo al VTAC para entrar en el modo de autocarga (vía PROM exterba).
1	0	0	0	Leer dirección de la línea del cursor	
1	0	0	1	Leer dirección de carácter del cursor	
1	0	1	0	Puesta a cero	Posiciona la cadena de tiempos en el borde superior izquierdo de la página. La posición se enclava en la pastilla mediante $\overline{DS}$ y los contadores se retienen hasta que son liberados por la orden de arranque.
1	0	1	1	Desplazamiento hacia arriba («scroll up») o enrollamiento ascendente	Incrementa la dirección de la primera fila de datos representada en la página; esto es, antes de la recepción de la orden del desplazamiento hacia arriba: línea superior = 0, línea inferior = 23. Tras recibir la orden de desplazamiento hacia arriba, línea superior = 1, línea inferior = 0.
1	1	0	0	Cargar dirección del carácter del cursor*	
1	1	0	1	Cargar dirección de la línea del cursor	
1	1	1	0	Arrancar cadena de tiempos	La recepción de esta orden después de otra de Puesta en Condiciones Iniciales o Autocarga por procesador libera la cadena de tiempos aproximadamente una línea de exploración más tarde. En las aplicaciones que requieren funcionamiento síncrono de más de un CRT 5027, el acarreo del contador de puntos debe mantenerse a nivel bajo durante el $\overline{DS}$ para esta orden.
1	1	1	1	Autocarga sin procesador	El dispositivo comenzará a cargarse mediante PROM cuando $\overline{DS}$ pase a nivel bajo. La orden 1111 debe mantenerse en A3-ø el tiempo suficiente para garantizar la autocarga. (El contador de exploraciones debe completar al menos un ciclo.) La autocarga se termina automáticamente y la cadena de tiempos arrancada cuando se retira la condición de todo «1», independientemente de $\overline{DS}$. Para funcionamiento síncrono de más de un VTAC, el arrastre del Contador de Puntos debe mantenerse a nivel bajo cuando se retira la orden.

*NOTA: Durante la autocarga, el Registro de Dirección del Carácter del Cursor (REG 7) y el de Dirección de Fila del Cursor (REG 6) están habilitados durante los estados ø111 y 1øøø de las salidas R3-Rø respectivamente del Contador de Exploraciones. Por tanto, los datos del cursor deben almacenarse en estas direcciones de la PROM.

TABLA 1

DIAGRAMA DE ASIGNACION DE BITS

DIAGRAMA DE TIEMPO EN C.A.

FIGURA 1. SINCRONIZACION EN VIDEO

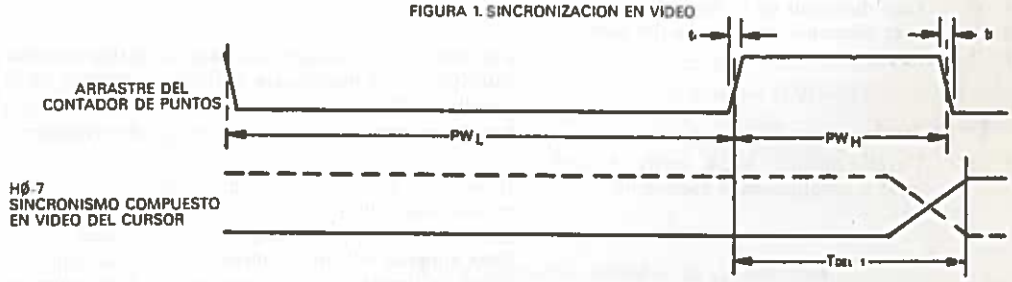


FIGURA 2. SINCRONIZACION CARGA/LECTURA

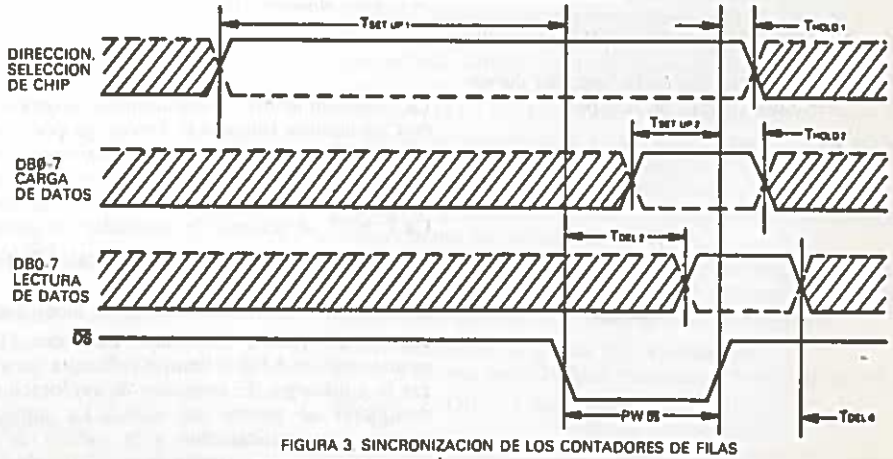
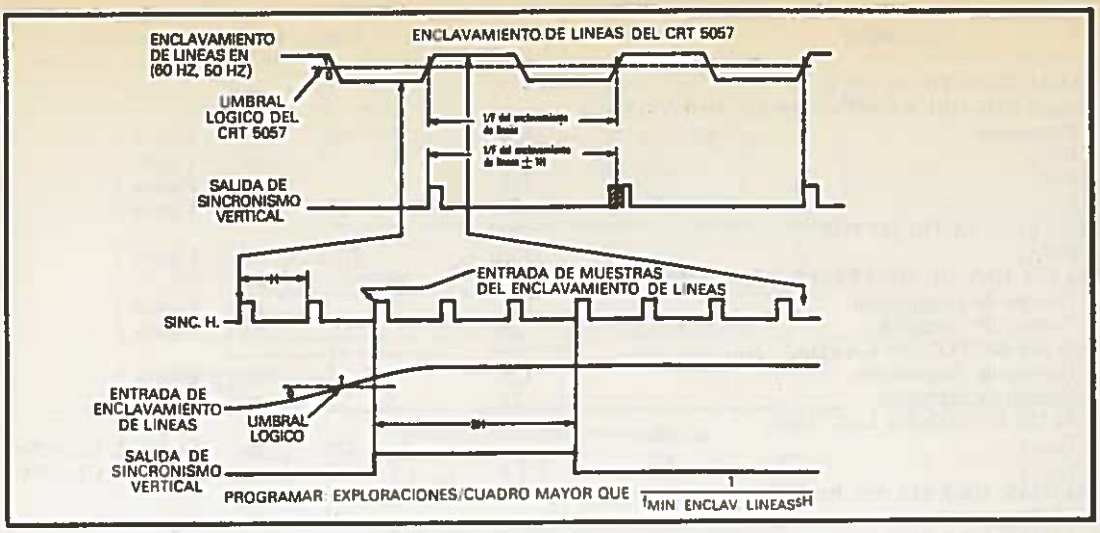


FIGURA 3. SINCRONIZACION DE LOS CONTADORES DE FILAS

*RØ-3 Y DRØ-5 pueden variar antes del flanco descendente del Sinc. H



REGIMENES MAXIMOS GARANTIZADOS*

Margen de temperatura de funcionamiento	0° a = +70 °C
Margen de temperatura de almacenamiento	−55 a +150 °C
Temperatura del estaño (soldadura en 10 seg)	+325 °C
Tensión positiva en cualquier patilla, V_{CC}	+18,0 V
Tensión negativa en cualquier patilla, V_{CC}	−0,3 V

* Los esfuerzos superiores a los citados con anterioridad pueden ocasionar daño irreparable al dispositivo. La presente es sólo una clasificación de sobrecarga y no implica el funcionamiento del dispositivo bajo estas u otras condiciones por encima de las indicadas en los apartados de funcionamiento de las hojas de características.

NOTA: Cuando se aplica tensión a este dispositivo desde una fuente de alimentación del laboratorio, o del sistema, es importante no exceder los Regímenes Máximos Garantizados, ya que ello puede conducir a un fallo del dispositivo. Algunas fuentes de alimentación presentan picos o fallos en su salida cuando se enchufa y desenchufa la c.a. Además, los transistores en la línea de c.a. pueden aparecer en la salida de c.c. Por ejemplo, la fuente de alimentación de banco programada para entregar 12 V puede mostrar grandes transistores de tensión cuando se la conecta y desconecta de la red. Si esta posibilidad existe, se recomienda emplear un circuito limitador de tensión.

CARACTERISTICAS ELECTRICAS (T_A = 0 °C a 70 °C, V_{CC} = +5 V ± 5 %, V_{DD} = +12 V ± 5 %, salvo que se especifique otra cosa.)

Parámetro	Mín.	Típ.	Máx.	Unidad	Comentarios
CARACTERISTICAS EN C.C.					
NIVELES DE TENSION DE ENTRADA					
Nivel bajo, V_{IL}			0,8	V	
Nivel alto, V_{IH}	V_{CC}-1,5		V_{CC}	V	
NIVELES DE TENSION DE SALIDA					
Nivel bajo, V_{OL} para RØ-3			0,4	V	I_{OL} = 3,2ma
Nivel bajo, V_{OL} todos los demás			0,4	V	I_{OL} = 1,6ma
Nivel alto, V_{OH} para RØ-3, DBØ-7	2,4				I_{OH} = 80μa
Nivel alto, V_{OH} todos los demás	2,4				I_{OH} = 40μa
CORRIENTE DE ENTRADA					
Nivel bajo, I_{IL} (Dirección, CS solamente)			250	μA	V_{IN} = 0,4 V
Fuga I_{IL} (Todas las entradas excepto dirección, CS)			10	μA	$O \leq V_{IN} \leq V_{CC}$
CAPACITANCIA DE ENTRADA					
Bus de datos, C_{IN}		10	15	pF	
Reloj $\overline{DS}$, C_{IN}		25	40	pF	
Todos los demás, C_{IN}		10	15	pF	
FUGA EN EL BUS DE DATOS EN MODO DE ENTRADA					
I_{DB}			10	μA	0,4 V ≤ V_{IN} ≤ 5,25 V
CORRIENTE DE ALIMENTACION					
I_{CC}		80	100	mA	
I_{DD}		40	70	mA	

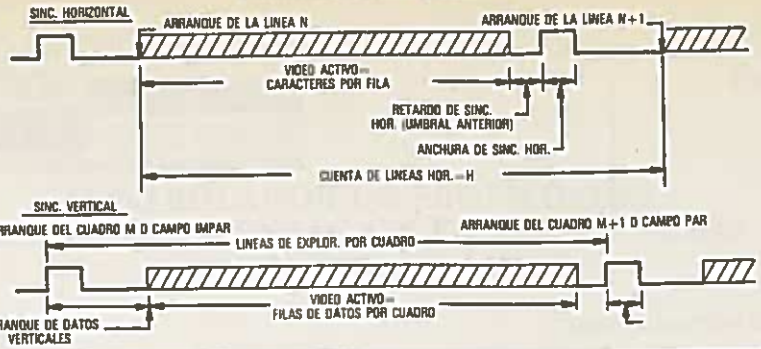
Parámetro	Mín.	Típ.	Máx.	Unidad	Comentarios
CARACTERISTICAS EN C.A.					$T_A = 25\,^{\circ}C$
ARRASTRE DEL CONTADOR DE PUNTOS					
Frecuencia	0,2		4,0	MHz	Figura 1
PW_H	35			ns	Figura 1
PW_L	215			ns	Figura 1
t_r, t_f		10	50	ns	Figura 1
REFERENCIA DE DATOS					
$PW_{\overline{DS}}$	150ns		10μs		Figura 2
SELECCION DE DIRECCION DE CHIP					
Tiempo de preparación	125			ns	Figura 2
Tiempo de retención	50			ns	Figura 2
BUS DE DATOS-EN CARGA					
Tiempo de preparación	125			ns	Figura 2
Tiempo de retención	75			ns	Figura 2
BUS DE DATOS-EN LECTURA					
T_{DEL2}			125	ns	Figura 2, CL = 50pF
T_{DEL4}	5		60	ns	Figura 2, CL = 50pF
SALIDAS: HØ-7, HS, VS, BL, CRV,					
$\overline{CS}$-T_{DEL1}			125	ns	Figura 1, CL = 20pF
SALIDAS: RØ-3, DRØ-5					
T_{DEL3}	*		500	ns	Figura 3, CL = 20pF

* RØ-3 y DRØ-5 pueden variar antes del flanco descendente del sincro H.

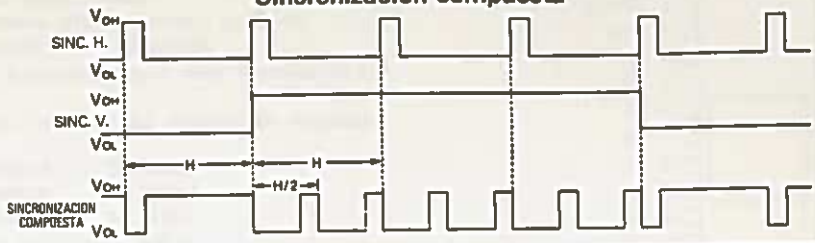
Restricciones

1. Sólo existe una patilla disponible para introducir datos en el dispositivo a través del bus de datos. Por tanto, las coordenadas X e Y del cursor se cargan en la patilla presentando un conjunto de direcciones y se sacan presentándole un conjunto nuevo de direcciones. Por ello, las señales de control WRITE (escritura) y READ (lectura) normalizadas de la mayoria de los microprocesadores han de controlarse mediante puerta NOR externa para ofrecer un único impulso de referencia al dispositivo.
2. En el modo entrelazado, el número total de huecos para caracteres asignado a la exploración horizontal ha de ser par con el fin de asegurar que el sincronismo vertical ocurra exactamente entre impulsos de sincronismo horizontal.

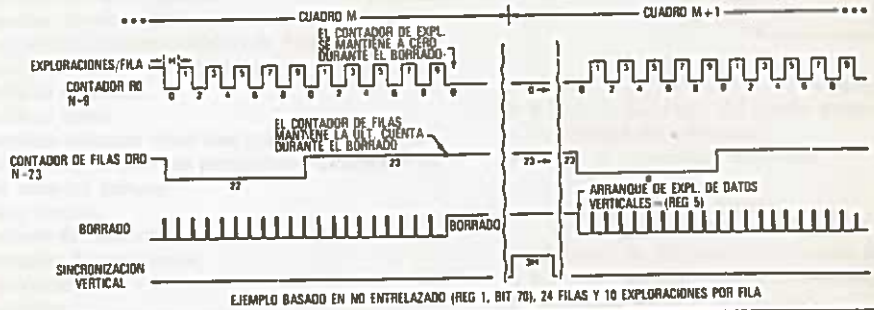
Sincronización general



Sincronización compuesta



Sincronización vertical



EJEMPLO BASADO EN NO ENTRELAZADO (REG 1, BIT 70), 24 FILAS Y 10 EXPLORACIONES POR FILA

Arranque del CRT 5027

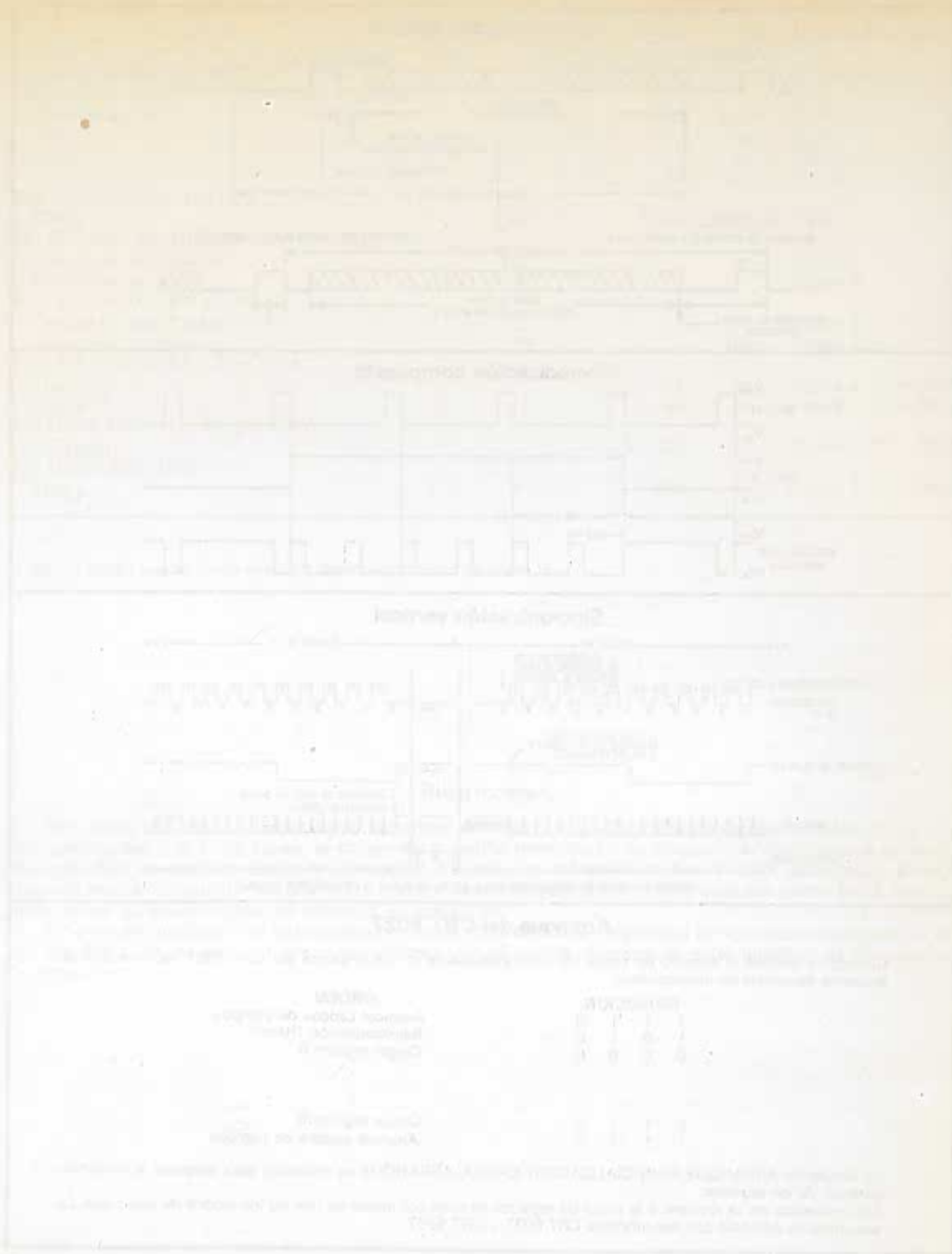
Cuando se emplea el método de carga por microprocesador de los registros del CRT 5027, es necesaria la siguiente secuencia de instrucciones:

DIRECCION	ORDEN
1 1 1 0	Arrancar cadena de tiempos
1 0 1 0	Reinicialización (Reset)
0 0 0 0	Cargar registro 0
⋮	⋮
0 1 1 0	Cargar registro 6
1 1 1 0	Arrancar cadena de tiempos

La secuencia ARRANQUE-REINICIALIZACION-CARGA-ARRANQUE es necesaria para asegurar la iniciación correcta de los registros.
Esta secuencia no se requiere si la carga de registros se hace por medio de uno de los modos de autocarga. La secuencia es opcional con los modelos CRT 5037 y CRT 5057.

STANDARD MICROSYSTEMS CORPORATION
We keep ahead of our competition so you can keep ahead of yours.

Los diagramas de circuitos que utilizan productos SMC se incluyen como medio de ilustración de las aplicaciones típicas de semiconductores; en consecuencia, no es necesario presentar una información completa y suficiente para las necesidades de diseño. La información ha sido comprobada cuidadosamente y se cree que es completamente fiable. Sin embargo, no se asume responsabilidad alguna por las inexactitudes que pudiera haber. Es más, tal información no cede al comprador del dispositivo semiconductor descrito permiso alguno sobre los derechos de patente de SMC u otros. SMC se reserva el derecho a introducir cambios en cualquier momento con el fin de mejorar el diseño y suministrar el mejor producto posible.



Apéndice C9

STANDARD MICROSYSTEMS CORPORATION
35 Marcus Blvd Hauppauge N Y 11787
(516) 273-3100 TWX 510-227-8898
We keep ahead of our competition so you can keep ahead of yours.

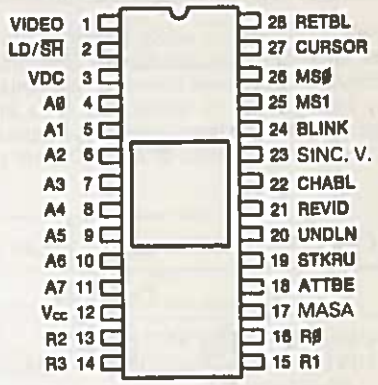
CRT 8002
Familia μPC

CONTROLADOR DE PROPIEDADES DE LA REPRESENTACION EN TRC DE VIDEO GENERADOR DE VIDEO VDAC

CARACTERISTICAS

- Generador de caracteres incorporado en la patilla (con máscara programable):
  128 caracteres (alfanuméricos y gráficos).
  Bloque matriz de 7 × 11 puntos.
- Registro de desplazamiento en vídeo incorporado en la patilla.
  Frecuencia máxima del registro de desplazamiento:
  CRT 8002A 20 MHz
  CRT 8002B 15 MHz
  CRT 8002C 10 MHz
  Tiempo de acceso 400ns
- Supresor de vídeo para retrazado horizontal y vertical incorporado en la patilla.
- No precisa circuito «descensor».
- Cuatro modos (entremezcables) de funcionamiento:
  Generador de caracteres interno.
  Gráficos en vídeo.
  Gráficos finos.
  Entradas externas (matrices/gráficos puntuales).
- Circuitería lógica para las propiedades incluida en la patilla carácter campo:
  Vídeo inverso.
  Borrado de caracteres.
  Parpadeo de caracteres.
  Subrayado.
  Corchetes.
- Cuatro modos de cursor incluidos en la patilla:
  Subrayado.
  Subrayado con parpadeo.
  Vídeo inverso.
  Vídeo inverso con parpadeo.

CONFIGURACION DE PATILLAS



- Ritmo de parpadeo de caracter programable.
- Ritmo de parpadeo del cursor programable.
- Posibilidad de subíndices.
- Juego de caracteres ampliable:
  Matrices externas.
  Alfanuméricos y gráficos.
  RAM, ROM y PROM.
- Memoria de dirección incorporada en la patilla.
- Memoria de propiedades incorporada en la patilla.
- Funcionamiento a +5 V.
- Compatible con TTL.
- Proceso COPLAMOS de puertas de silicio MOS de canal-N.
- Tecnología CLASP, opciones ROM.
- Compatible con el VTAC CRT 5027.

Descripción general

El controlador de propiedades de la representación en TRC de vídeo (VDAC) CRT 8002 de la SMC es un dispositivo MOS/LSI COPLAMOS de canal-N que utiliza la tecnología CLASP. Contiene una ROM de 7 × 11 × 128 generadora de caracteres, modo de gráficos anchos, modo de gráficos finos, modo con entradas externas, memoria de direcciones/datos para los caracteres, circuitería lógica para control de las propiedades del campo y los caracteres, memoria para las propiedades, cuatro modos de cursor, dos ritmos de parpadeo programables y registro de desplazamiento en vídeo a alta velocidad. El VDAC CRT 8002 es la patilla asociada del VTAC CRT 5027 de la SMC. Ambas patillas juntas incluyen toda la circuitería necesaria para la parte de representaciones de un terminal de vídeo con TRC.

La salida en vídeo del CRT 8002 puede conectarse directamente a la entrada de vídeo de un monitor con TRC. La salida de borrado de la CRT 5027 se puede conectar directamente a la entrada de supresión del haz para el retorno de la CRT 8002, para suministrar ambos borrados de retorno horizontal y vertical de la salida de vídeo.

Hay cuatro modos de cursor disponibles en el CRT 8002. Son: subrayado, subrayado con parpadeo, bloque de vídeo inverso y bloque de vídeo inverso con parpadeo. Cualquiera de ellas puede programarse mediante máscara como función del cursor. Existe una tasa de parpadeo del cursor que puede programarse por separado mediante máscara para suministrar un ritmo de parpadeo entre 15 y 1 Hz.

Las propiedades del CRT 8002 comprenden: vídeo inverso, borrado de caracteres, parpadeo, subrayado y corchetes. La tasa de parpadeo de los caracteres puede programarse mediante máscara entre 7,5 y 0,5 Hz y posee un ciclo útil de 75/25. El subrayado y los corchetes funcionan de forma parecida, pero son funciones controladas independientemente y pueden programarse mediante máscara a cualquier número de líneas de la traza y en cualquier posición del bloque de caracteres. Estas propiedades son comunes a todos los modos.

En el modo de gráficos gruesos, el CRT 8002 produce una entidad gráfica del tamaño de un bloque de caracteres. Esta entidad gráfica está compuesta por 8 partes, cada una de las cuales está asociada con un bit de un octeto para gráficos. Suministrando así 256 símbolos gráficos diferentes. De esta forma, el CRT 8002 puede producir bien un símbolo alfanumérico, o bien una entidad gráfica, dependiendo del modo seleccionado. El modo puede cambiarse de acuerdo con cada carácter.

Impreso con permiso de SMC © 1980.

El modo de gráficos finos permite al usuario crear dibujos de una sola línea y formas.
El modo externo permite al usuario ampliar el juego de caracteres de la ROM incluida en la patilla y la capacidad de gráficos interna, mediante la inserción de símbolos externos. Estos símbolos externos pueden provenir bien de la RAM, bien de la ROM o bien de la PROM.

REGIMENES MAXIMOS GARANTIZADOS*

Margen de temperatura de funcionamiento	de 0°C a +70°C
Margen de temperatura de almacenamiento	de −55°C a +150°C
Temperatura del estaño (soldadura, 10 sg)	+325°C
Tensión positiva en cualquier terminal con respecto a tierra	+8,0 V
Tensión negativa en cualquier terminal con respecto a tierra	0,3 V

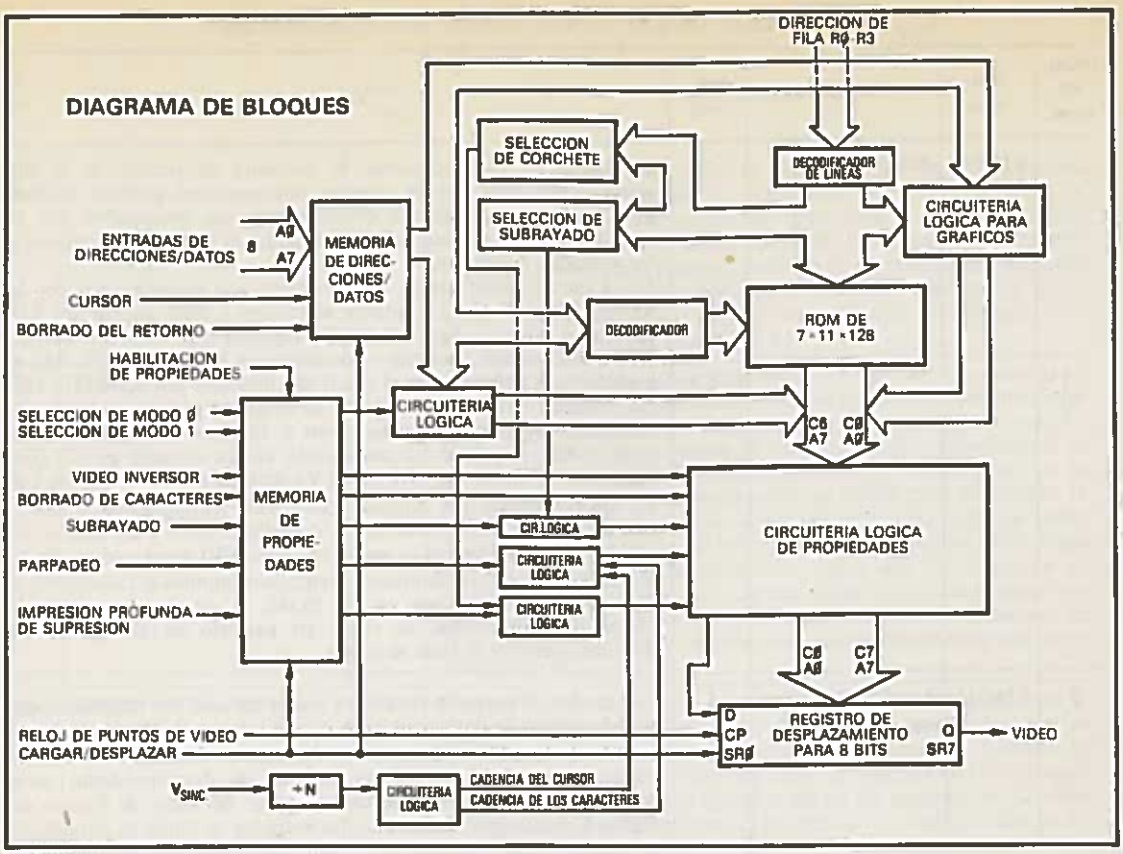
* Los esfuerzos superiores a los citados con anterioridad pueden ocasionar daño irreparable al dispositivo. La presente es sólo una clasificación de sobrecarga y no implica el funcionamiento del dispositivo bajo estas u otras condiciones por encima de las indicadas en los apartados de funcionamiento de las hojas de características.

NOTA: Cuando se aplica tensión a este dispositivo desde una fuente de alimentación del laboratorio o del sistema es importante no exceder los Regímenes Máximos Garantizados, ya que ello puede conducir a un fallo del dispositivo. Algunas fuentes de alimentación presentan picos o fallos en su salida cuando se enchufa y desenchufa la c.a. Además, los transistores en la línea de c.a. pueden aparecer en la salida de c.c. Por ejemplo, la fuente de alimentación de banco programada para entregar 12 V puede mostrar grandes transistores de tensión cuando se la conecta y desconecta de la red. Si esta posibilidad existe, se recomienda emplear un circuito limitador de tensión.

CARACTERISTICAS ELECTRICAS ($T_A = 0°$ a 70°C, $V_{CC} = \pm 5$ V ± 5 %, salvo que se indique otra cosa).

Parámetro	Mín.	Típ.	Máx.	Unid.	Comentarios
CARACTERISTICAS EN C.C.					
NIVELES DE TENSION DE ENTRADA					
Nivel bajo, V_{IL}			0,8	V	Excluyendo VDC
Nivel alto, V_{IH}	2,0			V	Excluyendo VDC
NIVELES DE TENSION DE RELOJ A LA ENTRADA					
Nivel bajo, V_{IL}			0,8	V	
Nivel alto, V_{IH}	4,3			V	Ver figura 6
NIVELES DE TENSION DE SALIDA					
Nivel bajo, V_{OL}			0,4	V	$I_{OL} = 0{,}4$ mA, 74LSXX carga
Nivel alto, V_{OH}	2,4			V	$I_{OH} = -20\,\mu A$
CORRIENTE DE ENTRADA					
Fuga, I_L (excepto reloj)			10	μA	$0 \leqslant V_{IN} \leqslant V_{CC}$
Fuga, I_L (solamente el reloj)			50	μA	$0 \leqslant V_{IN} \leqslant V_{CC}$
CAPACITANCIA DE ENTRADA					
Datos		10		pF	@ 1 MHz
LD/SH		20		pF	@ 1 MHz
Reloj		25		pF	@ 1 MHz
CORRIENTE DE ALIMENTACION					
I_{CC}		100		mA	
CARACTERISTICAS EN C.A.					
Ver figuras 6 y 7					

SIMBOLO	PARAMETRO	CRT 8002A		CRT 8002B		CRT 8002C		UNID.
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
VDC	Frecuencia del reloj de puntos del vídeo	1,0	20	1,0	15	1,0	10	MHz
PW_H	VDC-Tiempo alto	15,0		23		40		ns
PW_L	VDC-Tiempo bajo	15,0		23		40		ns
t_{CY}	LD/SH tiempo de ciclo	400		533		800		ns
t_r, t_f	Subir, tiempo de caída		10		10		10	ns
$t_{SET\text{-}UP}$	Entrada tiempo de establecimiento (set-up)	⩾0		⩾0		⩾0		ns
t_{HOLD}	Entrada tiempo de retención	15		15		15		ns
t_{PDL}, t_{PDO}	Salida retardo de propagación	15	50	15	65	15	100	ns
t_1	LD/SH tiempo de establecimiento	10		15		20		ns
t_2	LD/SH tiempo de retención	15		15		15		ns



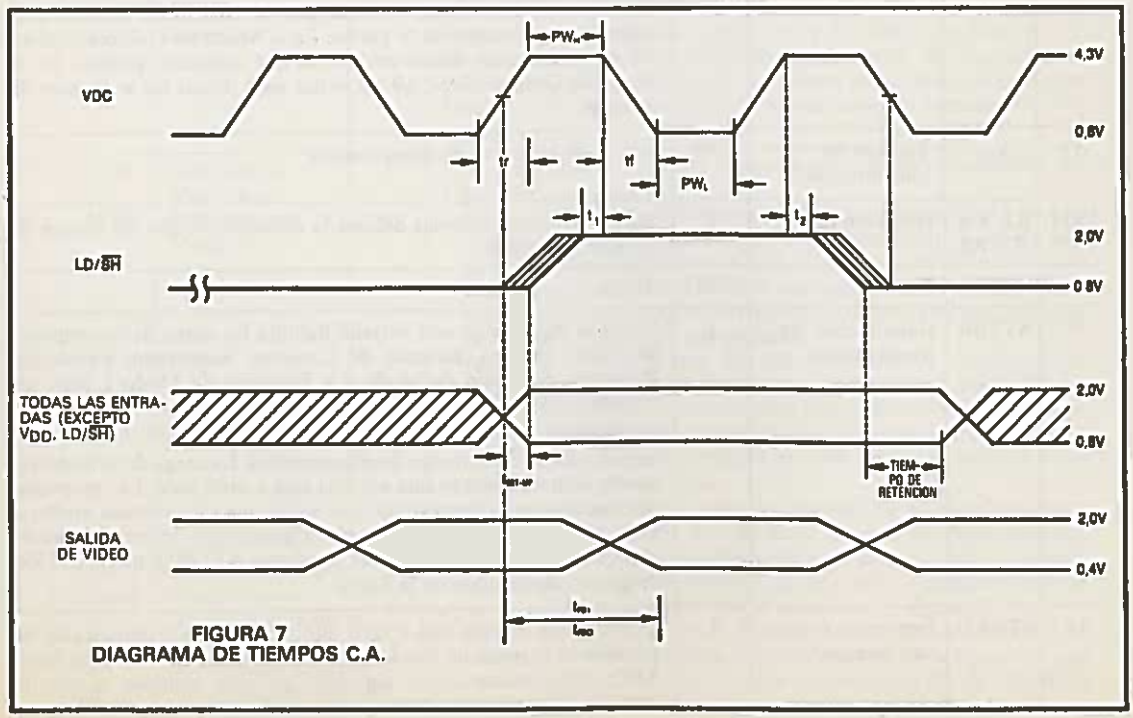


FIGURA 7
DIAGRAMA DE TIEMPOS C.A.

DESCRIPCION DE LAS FUNCIONES DE LOS TERMINALES

Núm. de term.	Símbolo	Nombre	Ent./ Sal.	Función
1	VIDEO	Salida de vídeo	0	La salida de vídeo contiene la corriente de puntos de la fila seleccionada bien por el carácter alfanumérico, gráficos anchos, gráficos finos o caracteres externos una vez procesados por la circuitería lógica de propiedades, así como el borrado del retorno y las entradas del cursor. En el modo alfanumérico, los caracteres son programados por la ROM en los 77 (7 × 11) puntos asignados a cada uno de los 128 caracteres. Véase la figura 5. La fila superior (RØ) y las que van de R12 a R15 suelen contener ceros como en la columna C7. Así, el carácter está definido por el cuadrado limitado por R1-R11 y CØ-C6. Cuando una fila de la ROM se carga en paralelo a través de circuitería lógica de propiedades, en el registro de desplazamiento para 8 bits el primer bit desplazado afuera en serie es C7. (Un «cero», o un «uno» en VIN INV.) Va seguido de C6, C5, hasta CØ. La sincronización del impulso Load/$\overline{\text{Shift}}$ (carga/desplazamiento) determinará el número de ceros de relleno (– –, de cero a N) adicionales (o «unos» si el modo es VID-INV) desplazados afuera. Véase la figura 4. Cuando aparece el siguiente impulso de Load/Shift, la fila del nuevo carácter en la ROM, a través de circuitería lógica de propiedades, se carga en paralelo en el registro de desplazamiento y el ciclo se repite.
2	LD/$\overline{\text{SH}}$	Load/$\overline{\text{Shif}}$ (carga/ desplazamiento)	I	Los modos de carga en paralelo y salida en serie por desplazamiento del registro de desplazamiento para 8 bits se establecen mediante la entrada Load/$\overline{\text{Shift}}$ (carga/desplazamiento). Cuando está a nivel bajo, esta entrada habilita el registro de desplazamiento para desplazar en serie un bit con cada impulso del Reloj de Puntos de Vídeo. Cuando está a nivel alto, las entradas de datos en paralelo (a lo ancho) quedan habilitadas y la carga síncrona ocurre al siguiente impulso del Reloj de Puntos de Vídeo. Durante la carga en paralelo se inhibe el flujo de datos en serie. Las entradas Direcciones/Datos (AØ-A7) se enclavan en la transición negativa de la entrada Load/$\overline{\text{Shift}}$ (carga/desplazamiento). Ver diagrama de tiempos en la figura 7.
3	VDC	Reloj de Puntos de Vídeo	I	Frecuencia a la que se desplaza el vídeo.
4-11	AØ-A7	Direcciones/Datos	I	En el Modo Alfanumérico, los 7 bits de las entradas AØ-A6 son decodificados internamente para que seleccionen uno de los 128 caracteres disponibles (A7 = X). En el Modo Externo, AØ-A7 sirve para insertar una palabra de 8 bits procedente de una ROM, PROM o RAM externa definida por el usuario en la circuitería lógica de propiedades de la patilla. En el Modo de Gráficos anchos, AØ-A7 se usa para definir una de las 256 entidades gráficas. En el Modo de Gráficos finos, AØ-A2 se usa para definir los segmentos de 3 líneas.
12	V_{CC}	Tensión de alimentación	PS	+5 V de la fuente de alimentación.
13,14, 15,16	R2, R3, R1, RØ	Dirección de fila	I	Estas 4 entradas binarias definen la dirección de fila del bloque de carácter en curso.
17	GND	Tierra	GND	Tierra.
18	ATTBE	Habilitación de propiedades	I	Un nivel positivo en esta entrada habilita los datos de las entradas de Vídeo Inverso, Borrado de Carácter, Subrayado, Paréntesis, Parpadeo, Selección de Modo Ø y Selección de Modo 1 para ser introducidos por el impulso de referencia en la memoria de propiedades de la patilla al tiempo de la transición negativa del impulso Load/$\overline{\text{Shift}}$ (carga desplazamiento). La carga de la memoria queda inhibida cuando esta entrada está a nivel bajo. Las propiedades enclavadas permanecerán fijas hasta que esta entrada vuelva a estar a nivel alto. Para facilitar el enclavamiento de propiedades en el modo de carácter por carácter, conéctese ATTBE a nivel alto. Ver diagrama de tiempos en la figura 7.
19	STKRU	Impresión profunda (tachado)	I	Cuando esta entrada está a nivel alto y RETBL = 0, las entradas en paralelo al registro de desplazamiento se fuerzan a nivel alto (SRØ-SR7), proporcionando un segmento de línea continua a todo lo largo del bloque de caracteres. Su direccionamiento se modifica con

DESCRIPCION DE LAS FUNCIONES DE LOS TERMINALES (Continuación)

Núm. de term.	Símbolo	Nombre	Ent./ Sal.	Función
19 (cont.)				el Vídeo Inverso (ver tabla 1). Además, hay en la patilla un decodificador programable en ROM para decodificar la cuenta de líneas en la que se va a colocar, así como para programar su altura para que sea de 1 a N líneas de traza. Realmente, la circuitería lógica del decodificador correspondiente (programable por máscara) permite que sea una distribución o cualquier número de líneas horizontales en el bloque de caracteres. El tipo normalizado será una línea doble en las filas R5 y R6.
20	UNDLN	Subrayado	I	Cuando esta entrada está a nivel alto y RETBL = 0, las entradas en paralelo al registro de desplazamiento se fuerzan a nivel alto (SRØ-SR7), proporcionando un segmento de línea continua a todo lo largo del bloque de caracteres. El funcionamiento del subrayado se modifica con el Vídeo Inversor (ver tabla 1). Además, hay en la patilla un decodificador programable en ROM para decodificar la cuenta de líneas en la que se va a colocar el subrayado, así como para programar la altura del subrayado para que sea de 1 a N líneas de traza. Realmente, la circuitería lógica del decodificador de subrayado (programable por máscara) permite que el subrayado sea un número cualquiera o una distribución de líneas horizontales en el bloque de caracteres. El subrayado normalizado será una línea sencilla en R11.
21	REVID	Vídeo Inverso	I	Cuando esta entrada está a nivel bajo y RETBL = 0, los datos de la circuitería lógica de propiedades se presenta directamente en las entradas en paralelo del registro de desplazamiento. Cuando el Vídeo Inverso está a nivel alto los datos se invierten en la circuitería lógica de propiedades y se presenta así en las entradas en paralelo del registro de desplazamiento. Esto invierte los datos y el campo de vídeo. Ver tabla 1.
22	CHABL	Borrado de caracteres	I	Cuando esta entrada se halla a nivel alto, las entradas en paralelo al registro de desplazamiento se llevan a nivel bajo, suministrando un segmento de línea con caracteres en blanco. El Borrado de Caracteres anula el parpadeo. El funcionamiento del Borrado de Caracteres es modificado por la entrada de Vídeo Inverso. Ver tabla 1.
23	V SINC	V SINC	I	Esta entrada se toma como entrada de reloj para los dos divisores de ritmo de parpadeo programables por máscara que hay en la patilla. El ritmo de parpadeo del cursor (50/50 de ciclo de trabajo) es dos veces el de los caracteres (75/25 de ciclo de trabajo). Los divisores se pueden programar ÷4 y ÷30 para el cursor (÷8 y ÷60 para los caracteres)
24	BLINK	Parpadeo	I	Cuando esta entrada está a nivel alto y RETBL = 0 y CHABL = 0, el carácter parpadeará al ritmo programado. El parpadeo se produce borrando el bloque de caracteres con el reloj interno para Parpadeo de Caracteres. El ritmo normalizado de parpadeo es 1875 hercios.
25 26	MS1 MSØ	Selección de Modo 1 Selección de Modo 2	I I	Estas entradas definen los cuatro modos de funcionamiento del CRT 8002 como sigue: *Modo Alfanumérico.* En este modo las direcciones AØ-A6 (A = X) son decodificadas internamente para seleccionar 1 de los 128 caracteres disponibles de la ROM. El carácter seleccionado, junto con la fila decodificada, definen una salida de 7 bits de la ROM, que ha de ser cargada en el registro de desplazamiento mediante la circuitería lógica de propiedades. *Modo de Gráficos Finos.* En este modo AØ-A2 (A3-A7 = X) se cargan en la circuitería lógica para gráficos finos junto con las direcciones de filas. Esta circuitería lógica define los segmentos de una entidad gráfica tal y como se describe en la figura 2. La parte superior de la entidad comienza en la fila 0000 y termina en la fila programable por máscara. *Modo Externo.* En este modo las entradas AØ-A7 van directamente de la memoria de caracteres al registro de desplazamiento mediante la circuitería lógica de propiedades. Así, el usuario puede definir matrices externas de caracteres o entidades gráficas de una PROM, ROM o RAM exterior. Ver figura 3. *Modo de Gráficos Anchos.* En este modo las entradas AØ-A7 definen una entidad gráfica como la descrita en la figura 1. Cada línea de la entidad gráfica viene determinada por la circuitería lógica de gráficos anchos de conjunción con las entradas de filas

MS1	MSØ	MODO
1	1	Alfanuméricos
1	0	Gráficos finos
0	1	Modo Externo
0	0	Gráficos gruesos

DESCRIPCION DE LAS FUNCIONES DE LOS TERMINALES (Continuación)

Núm. de term.	Símbolo	Nombre	Ent./ Sal.	Función
25 26 (cont.)				RØ a R3. En este modo cada segmento de la entidad está definido por uno de los bits de la palabra de 8 bits. Por tanto, los 8 bits pueden definir cualquiera de las 256 entidades gráficas posibles. Estas entidades pueden colindar entre sí formando una pauta continua o pueden estar entremezcladas con caracteres alfanuméricos. Cada entidad ocupa el espacio de un bloque de caracteres y, por tanto, requiere un octeto de memoria. Estos cuatro modos pueden entremezclarse de acuerdo con cada carácter.
27	CURSOR	Cursor	I	Cuando esta salida está habilitada se activa uno de los cuatro modos preprogramables del cursor. El modo del cursor es programable por máscara en la patilla. El cursor estándar será un bloque de vídeo inverso parpadeante (a 3,75 Hz). Los cuatro modos del cursor son: Subrayado: En este modo ocurre un subrayado (de 1 a N líneas de traza) en la posición programada al respecto. Subrayado con parpadeo: En este modo lo subrayado parpadea al ritmo del cursor. Bloque de Vídeo Inverso: En este modo el Bloque de Caracteres se dispone en Video Inverso. Bloque de Vídeo Inverso con parpadeo: En este modo el bloque de caracteres se dispone en Vídeo Inverso al ritmo de parpadeo del cursor. El bloque de caracteres alternará entre vídeo normal e inverso. Las funciones del cursor están listadas en la tabla 1.
28	RETBL	Borrado del retorno (retraza)	I	Cuando esta entrada se enclava a nivel alto, las entradas en paralelo al registro de desplazamiento resultan borradas incondicionalmente y se introducen ceros en el siguiente impulso Load/Shift (carga/desplazamiento). Esto borra el vídeo, independientemente de sus propiedades, durante el tiempo de los retrocesos vertical y horizontal.

TABLA 1

CURSOR	RETBL	REVID	CHABL	UNDLN*	FUNCION
X	1	X	X	X	«0» S.R. Todos
0	0	0	0	0	D (S.R.) Todos
0	0	0	0	1	«1» (S.R.)*
					D (S.R.) Todos los demás
0	0	0	1	X	«0» (S.R.) Todos
0	0	1	0	0	D (S.R.) Todos
0	0	1	0	1	«0» (S.R.)*
					D (S.R.) Todos los demás
0	0	1	1	X	«1» (S.R.) Todos
Subrayado*	0	0	0	X	«1» (S.R.)*
					D (S.R.) Todos los demás
Subrayado*	0	0	1	X	«1» (S.R.)*
					«0» (S.R.) Todos los demás
Subrayado*	0	1	0	X	«0» (S.R.)*
					D (S.R.) Todos los demás
Subrayado*	0	1	1	X	«0» (S.R.)*
					«1» (S.R.) Todos los demás
Parpadeo** Subrayado*	0	0	0	X	«1» (S.R.) Parpadeo
					D (S.R.) Todos los demás
Parpadeo** Subrayado*	0	0	1	X	«1» (S.R.) Parpadeo
					«0» (S.R.) Todos los demás
Parpadeo** Subrayado*	0	1	0	X	«0» (S.R.)* Parpadeo
					D (S.R.) Todos los demás
Parpadeo** Subrayado*	0	1	1	X	«0» (S.R.) *Parpadeo
					«1» (S.R.) Todos los demás

TABLA 1 (Continuación)

CURSOR	RETBL	REVID	GHABL	UNDLN*	FUNCION
Bloque VIDIN (vídeo inverso)	0	0	0	0	D̄ (S.R.) Todos
Bloque VIDIN (vídeo inverso)	0	0	0	1	«0» (S.R.)*
					D̄ (S.R.) Todos los demás
Bloque VIDIN (vídeo inverso)	0	0	1	X	«1» (S.R.) Todos
Bloque VIDIN (vídeo inverso)	0	0	0	1	«0» (S.R.)*
					D̄ (S.R.) Todos los demás
Bloque VIDIN (vídeo inverso)	0	1	0	0	D (S.R.) Todos
Bloque VIDIN (vídeo inverso)	0	1	0	1	«1» (S.R.)*
					D (S.R.) Todos los demás
Bloque VIDIN (vídeo inverso)	0	1	1	X	«0» (S.R.) Todos
Parpadeo** Bloque VIDIN	0	0	0	0	Vídeo normal/VIDIN al ritmo del parpadeo del cursor
Parpadeo** Bloque VIDIN	0	0	0	1	
Parpadeo** Bloque VIDIN	0	0	1	X	
Parpadeo** Bloque VIDIN	0	1	0	0	
Parpadeo** Bloque VIDIN	0	1	0	1	
Parpadeo** Bloque VIDIN	0	1	1	X	

* En la decodificación de fila seleccionada.
** Al ritmo de parpadeo del cursor.

Nota: Si un carácter está parpadeando al ritmo asignado para los caracteres, el cursor lo cambiará al ritmo del cursor.

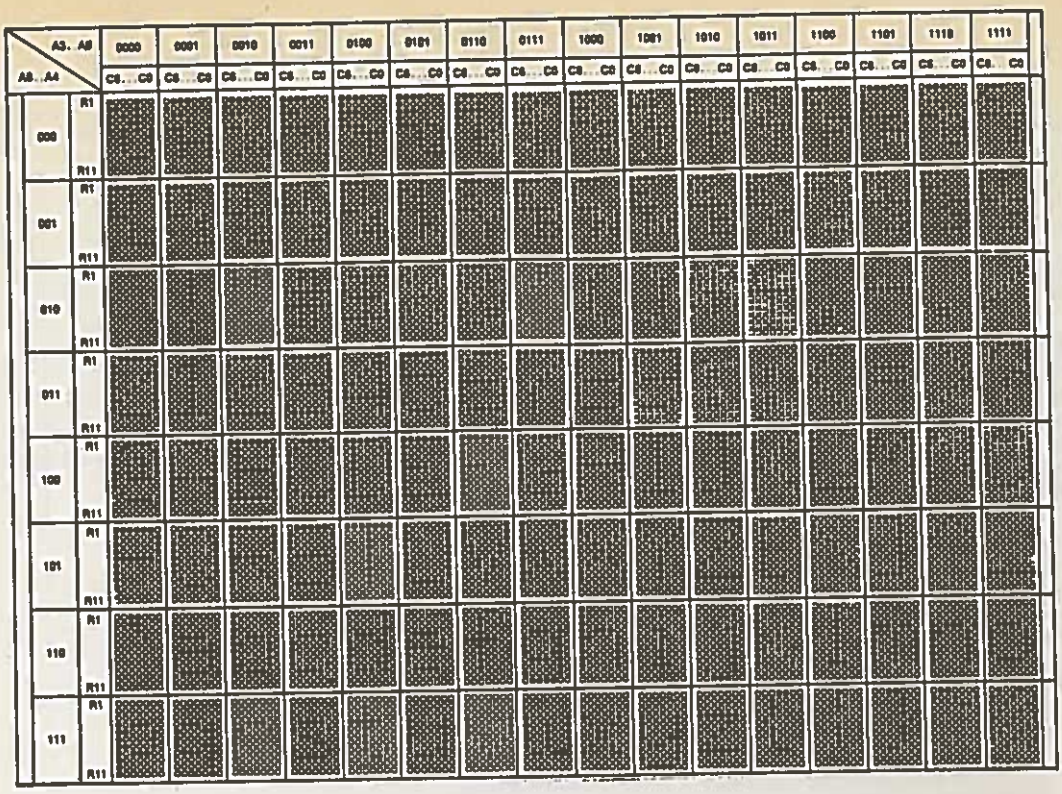
FIGURA 5

FORMATO DEL BLOQUE DE CARACTERES EN ROM

											FILAS	R3	R2	R1	RØ
(TODOS CEROS) →	0	0	0	0	0	0	0	0	–	–	RØ	0	0	0	0
	0	0	0	0	0	0	0	0	–	–	R1	0	0	0	1
	0	0	0	0	0	0	0	0	–	–	R2	0	0	1	0
	0	0	0	0	0	0	0	0	–	–	R3	0	0	1	1
	0	0	0	0	0	0	0	0	–	–	R4	0	1	0	0
	0	0	0	0	0	0	0	0	–	–	R5	0	1	0	1
77 BITS (7 × 11 ROM)	0	0	0	0	0	0	0	0	–	–	R6	0	1	1	0
	0	0	0	0	0	0	0	0	–	–	R7	0	1	1	1
	0	0	0	0	0	0	0	0	–	–	R8	1	0	0	0
	0	0	0	0	0	0	0	0	–	–	R9	1	0	0	1
	0	0	0	0	0	0	0	0	–	–	R1Ø	1	0	1	0
	0	0	0	0	0	0	0	0	–	–	R11	1	0	1	1
	0	0	0	0	0	0	0	0	–	–	R12	1	1	0	0
	0	0	0	0	0	0	0	0	–	–	R13	1	1	0	1
(TODOS CEROS)	0	0	0	0	0	0	0	0	–	–	R14	1	1	1	0
	0	0	0	0	0	0	0	0	–	–	R15	1	1	1	1
	*C7	C6	C5	C4	C3	C2	C1	CØ	–	–					

*LA COLUMNA 7 CONTIENE TODOS CEROS (VIDIN = 0)
LA COLUMNA 7 SE DESPLAZA PRIMERAMENTE

CEROS PROPAGADOS (DE RELLENO) PARA ESPACIAR LOS CARACTERES (NUMERO CONTROLADO POR LD/SH, SINCRONIZACION VDC)



CONSULTESE A FABRICA PARA MATRICES ESPECIALES Y OPCIONES PARA FORMATOS DE PROPAGACION

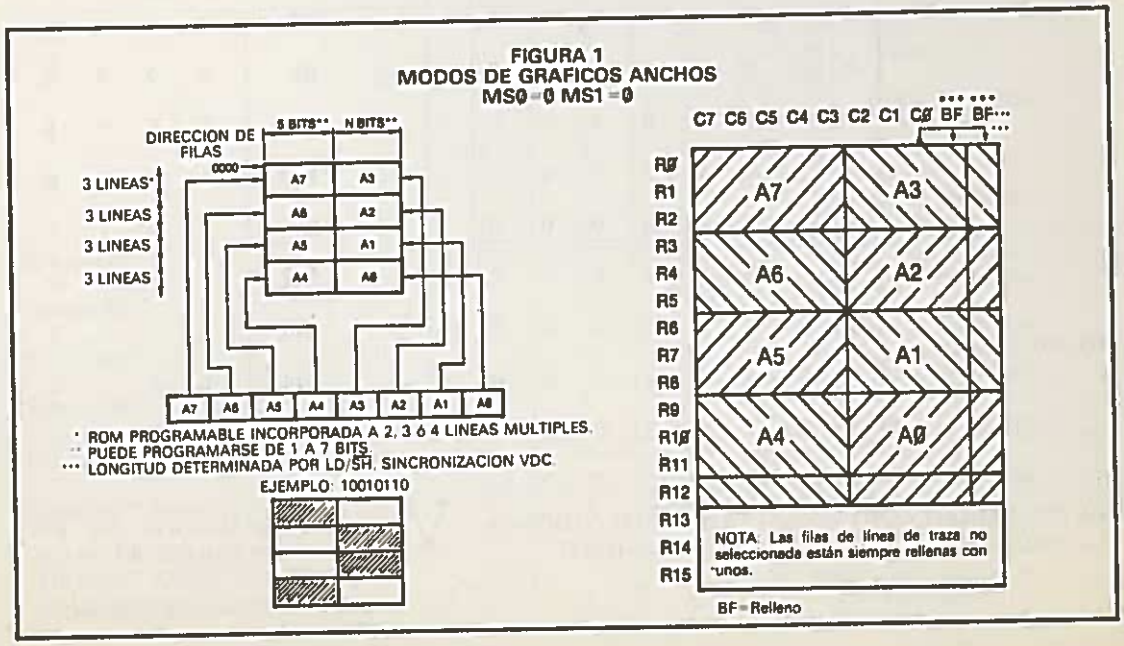
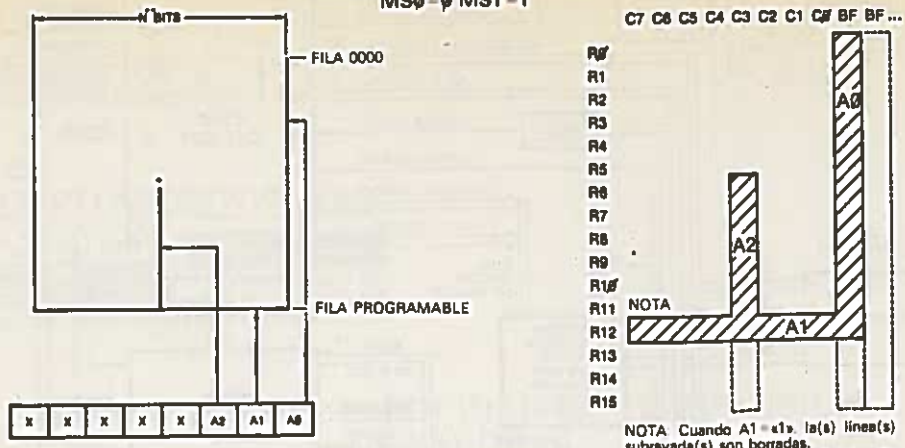


FIGURA 2
MODO DE GRAFICOS FINOS
MSØ = Ø MS1 = 1



X = IRRELEVANTE
* EL SEGMENTO INTERIOR ES PROGRAMABLE POR MASCARA A LA FILA 0000
** LONGITUD DETERMINADA POR LD/SH. SINCRONIZACION VDC

NOTA Cuando A1 = «1», la(s) línea(s) subrayada(s) son borradas. Cuando A1 = «0», el subrayado, si se lo selecciona, aparece.

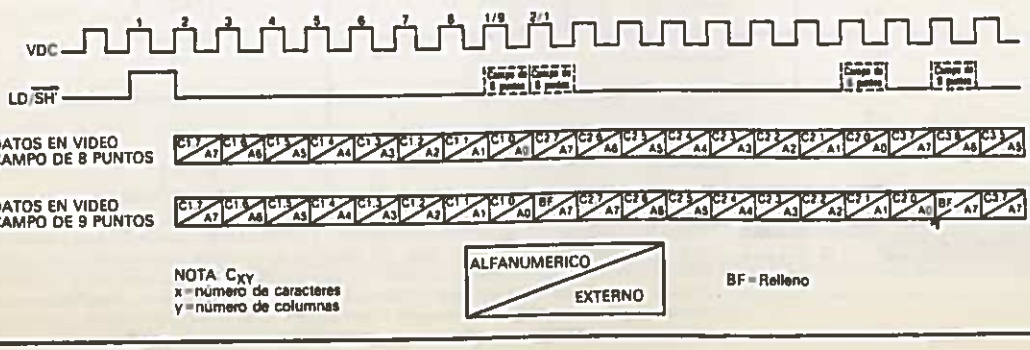
BF = Relleno

FIGURA 3
MODO EXTERNO
MSØ = 1 MS1 = Ø

	C7	C6	C5	C4	C3	C2	C1	CØ	BF	BF	···
RØ – R15	A7	A6	A5	A4	A3*	A2	A1	AØ	A7	A7	···

BF = Relleno

FIGURA 4. SALIDA TIPICA DE VIDEO



NOTA C_{XY}
x = número de caracteres
y = número de columnas

BF = Relleno

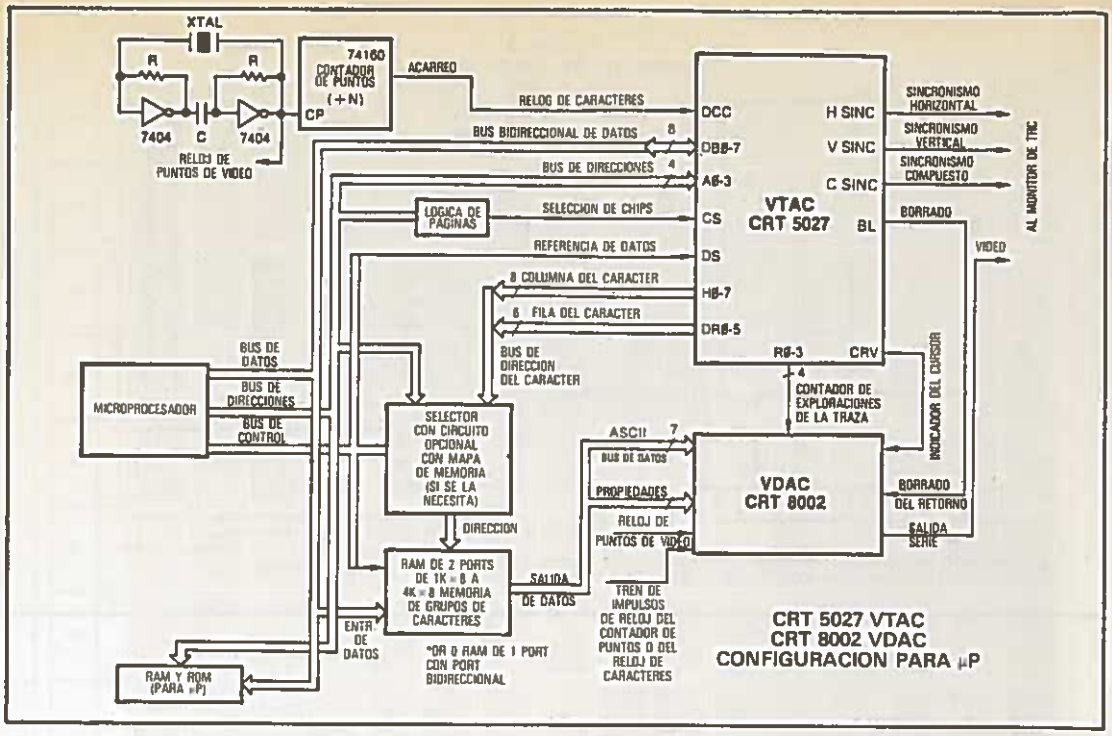
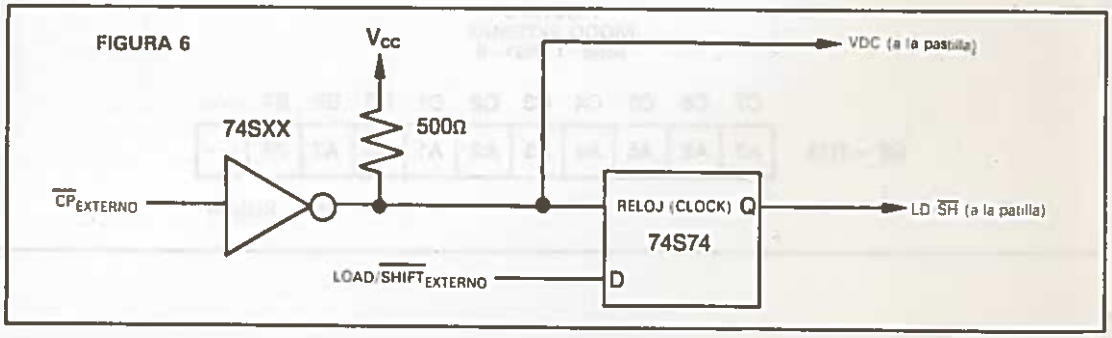


FIGURA 6



STANDARD MICROSYSTEMS CORPORATION

We keep ahead of our competition so you can keep ahead of yours.

Los diagramas de circuitos que utilizan productos SMC se incluyen como medio de ilustración de las aplicaciones típicas de semiconductores; en consecuencia, no es necesario presentar una información completa y suficiente para las necesidades de diseño. La información ha sido comprobada cuidadosamente y se cree que es completamente fiable. Sin embargo, no se asume responsabilidad alguna por las inexactitudes que pudiera haber. Es más, tal información no cede al comprador el dispositivo semiconductor descrito permiso alguno sobre los derechos de patente de SMC u otros. SMC se reserva el derecho a introducir cambios en cualquier momento con el fin de mejorar el diseño y suministrar el mejor producto posible.

Apéndice C10

STANDARD MICROSYSTEMS CORPORATION
35 Marcus Blvd Hauppauge N.Y. 11787
(516) 273-3100 TWX 510-227-8898
We keep ahead of our competition so you can keep ahead of yours.

COM 8046
COM 8046T

GENERADOR DE VELOCIDAD DE TRANSMISION EN BAUDIOS
Divisor programable

CARACTERISTICAS

- Oscilador a cristal incorporado o entrada de frecuencia exterior.
- Alimentación a +5 V únicos.
- Elección entre 32 frecuencias de salida.
- 32 regímenes asíncronos/síncronos de baudios.
- Compatibilidad directa con UART/USRT/ASTRO/USYNRT.
- ROM reprogramable mediante tecnología CLASP, lo que permite la generación de otras frecuencias.
- Compatible con TTL y MOS.
- Reloj 1X mediante patilla $f_o/16$.
- Salida para la frecuencia de cristal mediante salidas f_x y $f_x/4$.
- Inhabilitación de la salida mediante FENA.

CONFIGURACION DE TERMINALES

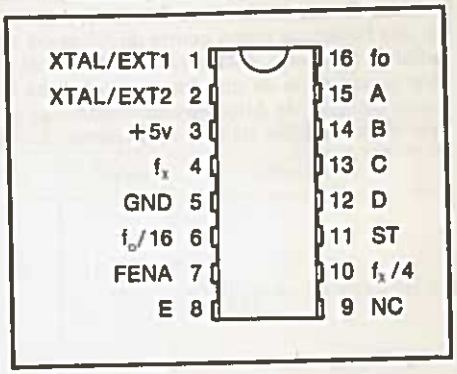
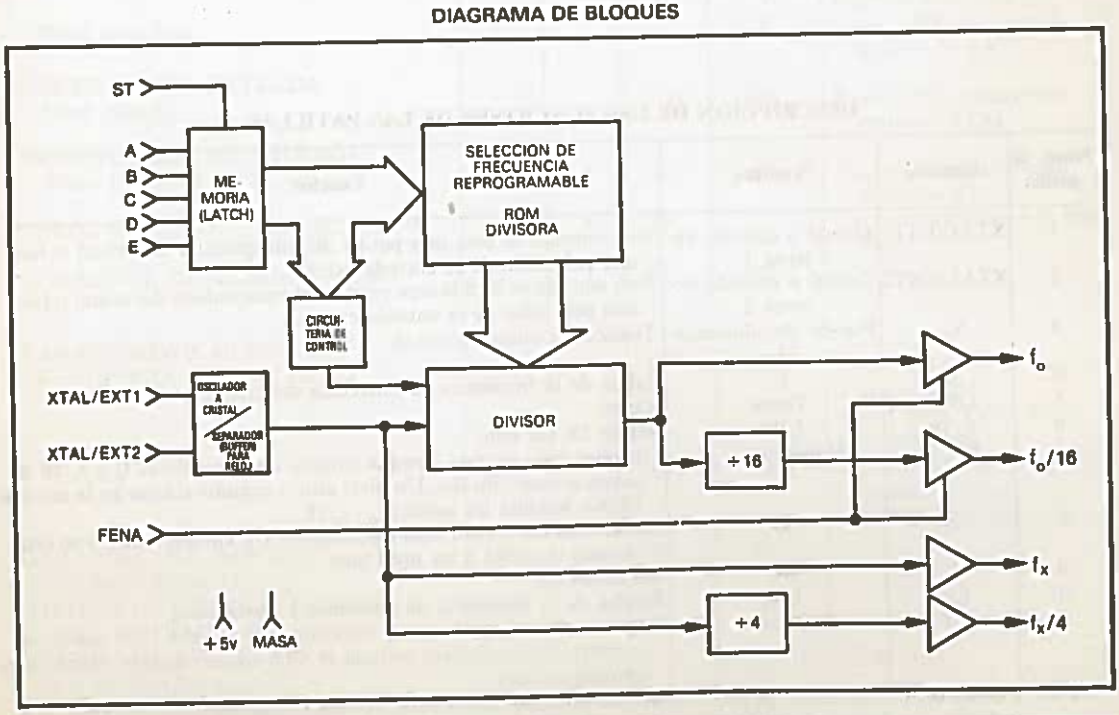


DIAGRAMA DE BLOQUES



Impreso con permiso de SMC © 1980.

Descripción general

El COM 8046 de la Standard Microsystems es una versión mejorada del Generador de Régimen de Baudios COM 5046. Está fabricado según los procesos COMPLAMOS y CLASP patentados por la SMC y emplea cargas en el modo de empobrecimiento que le permiten funcionar con una sola alimentación de +5 V.

El COM 8046 normalizado está especialmente destinado a generar un espectro completo de frecuencias para comunicaciones asíncronas/síncronas de datos para dispositivos de 1X, 16X y 32X, UART/USRT/ASTRO/USYNRT.

El COM 8046 ofrece un oscilador a cristal interno que se puede utilizar para suministrar la frecuencia maestra de referencia. Pero como alternativa, también se le puede suministrar una referencia externa, aplicándole señales de nivel TTL complementario a los terminales 1 y 2. Los modelos marcados 8046T son utilizables únicamente con referencias externas TTL. Las salidas de TTL que se emplean para alimentar los COM 8046 y 8046T no deben usarse para alimentar otras entradas TTL, so pena de comprometer su inmunidad al ruido como perjuicio de la excesiva carga.

La frecuencia de referencia (f_x) se usa para proporcionar dos salidas de frecuencia alta: una a f_x y otra a $f_x/4$. La salida de $f_x/4$ alimenta una carga tipo 7400, mientras que la f_x lo hace con dos cargas 74LS.

La salida del oscilador/separador se aplica al divisor para la generación de la frecuencia de salida f_o. El divisor es capaz de dividir por cualquier entero entre 6 y $2^{19}+1$, ambos inclusive. Si el divisor es par, la salida será cuadrada, de otro modo la salida estará a nivel alto más tiempo que a nivel bajo en la magnitud de un periodo de reloj f_x. La salida del divisor está también dividida internamente por 16 y disponible en la patilla de salida f_o y $f_o/16$. Esta salida puede alimentar una y la salida f_o dos cargas normalizadas TTL 7400. Ambas salidas f_o y $f_o/16$ pueden inhibirse suministrando un nivel lógico bajo al terminal de entrada FENA. Nótese que la entrada FENA posee una resistencia de acoplamiento al positivo que daría lugar a que la tensión en el terminal ascendiese al valor aproximado de V_{CC}, si se la deja desconectada.

La ROM divisora contiene 32 divisores, cada uno de 19 bits de anchura, y está fabricada empleando el proceso CLASP patentado de la SMC. Ello permite la reducción del tiempo de reciclaje para los patrones de la ROM. Los 5 bits de selección de divisores se retienen en una memoria (latch) de referencia externa. La entrada de este impulso de referencia es sensible al nivel: mientras es alto, los datos pasan directamente por la ROM. La iniciación de una frecuencia nueva ocurre de 3,5 μs del cambio de alguno de los 5 bits de selección del divisor y no se requiere señal de referencia. Esta característica puede inhibirse mediante una opción de programación CLASP que origina que la iniciación de una frecuencia distinta se retrase hasta que se completa un semiciclo de la f_o en curso. Las cinco entradas de datos poseen resistencias de polarización (pull-up) idénticas a la de la entrada FENA, mientras que la del impulso de referencia carece de ella.

DESCRIPCION DE LAS FUNCIONES DE LAS PATILLAS

Núm. de patilla	Símbolo	Nombre	Función
1	XTAL/EXT1	Cristal o entrada externa 1	Esta entrada es bien una patilla del encapsulado del cristal o bien una polaridad de la entrada externa.
2	XTAL/EXT2	Cristal o entrada externa 2	Esta entrada es bien la otra patilla del encapsulado del cristal o bien una polaridad de la entrada externa.
3	V_{CC}	Fuente de alimentación	Tensión de alimentación de +5 V.
4	f_x	f_x	Salida de la frecuencia de referencia del cristal/reloj.
5	GND	Tierra	Tierra.
6	$f_o/16$	$f_o/16$	Salida 1X del reloj.
7	FENA	Habilitación	Un nivel bajo en esta entrada origina que las salidas f_o y $f_o/16$ se vayan a nivel alto fijo. Un nivel alto o circuito abierto en la entrada FENA habilita las salidas f_o y $f_o/16$.
8	E	E	Bit selector del divisor más significativo. Un circuito abierto en esta entrada equivale a un nivel alto.
9	NC	NC	Sin conexión.
10	$f_x/4$	$f_x/4$	Salidas de la frecuencia de referencia ¼ cristal/reloj.
11	ST	Referencia	Impulso de referencia para selección del divisor. Los datos se muestrean cuando esta entrada es alta, conservándose cuando esta entrada es baja.
12-15	D, C, B, A	D, C, B, A	Bits de selección del divisor. A = bit menos significativo (LSB). Un circuito abierto en estas entradas equivale a un nivel alto.
16	f_o	f_o	Salida de reloj 16X.

CARACTERISTICAS ELECTRICAS COM 8046, COM 8046T, COM 8116, COM 8116T, COM 8126, COM 8126T, COM 8136, COM 8136T, COM 8146, COM 8146T

REGIMENES MAXIMOS GARANTIZADOS*

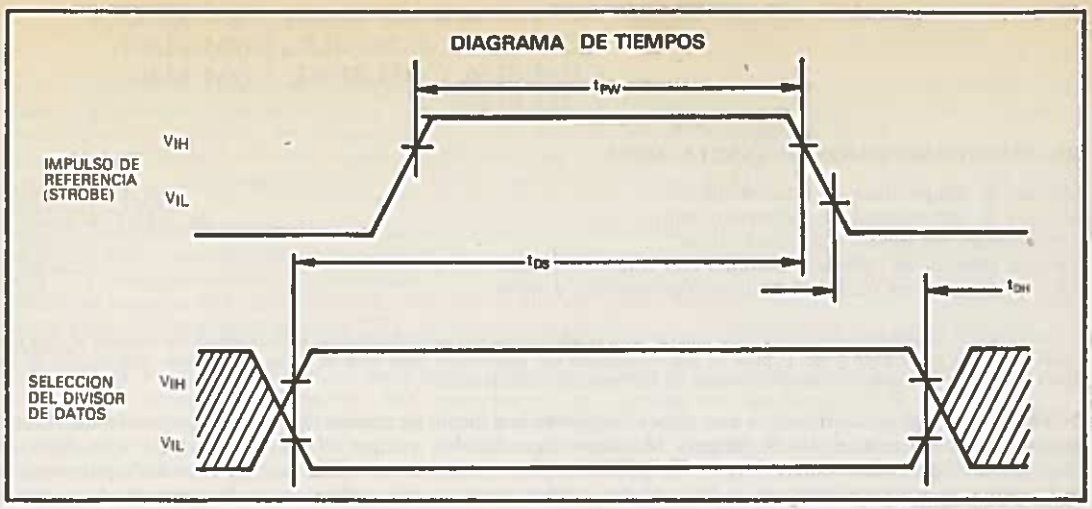
Margen de temperatura de funcionamiento	de 0 °C a +70 °C
Margen de temperatura de almacenamiento	de −55 °C a +150 °C
Temperatura del estaño (soldadura, 10 s)	+325 °C
Tensión positiva en cualquier terminal con respecto a tierra	+8,0 V
Tensión negativa en cualquier terminal con respecto a tierra	−0,3 V

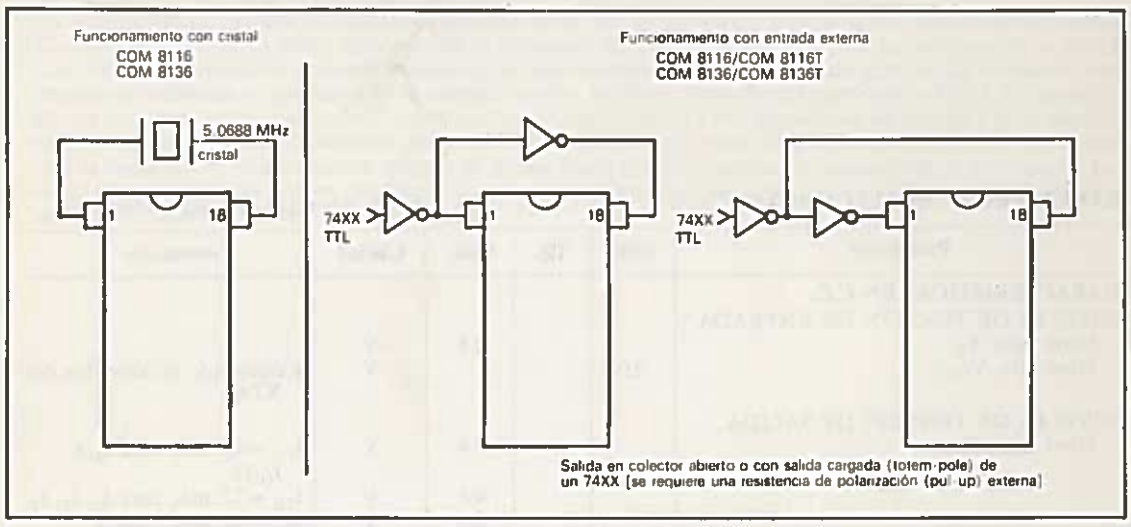
* Los esfuerzos superiores a los citados con anterioridad pueden ocasionar daño irreparable al dispositivo. La presente es sólo una clasificación de sobrecarga y no implica el funcionamiento del dispositivo bajo estas u otras condiciones por encima de las indicadas en los apartados de funcionamiento de las hojas de características.

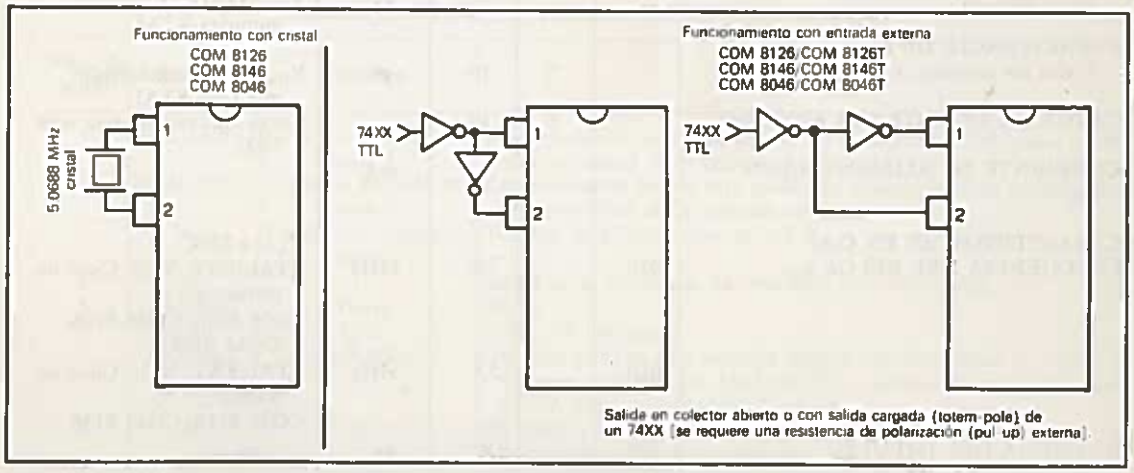
NOTA: Cuando se aplica tensión a este dispositivo desde una fuente de alimentación del laboratorio o del sistema es importante no exceder los Regímenes Máximos Garantizados, ya que ello puede conducir a un fallo del dispositivo. Algunas fuentes de alimentación presentan picos o fallos en su salida cuando se enchufa y desenchufa la c.a. Además, los transistores en la línea de c.a. pueden aparecer en la salida de c.c. Por ejemplo, la fuente de alimentación de banco programada para entregar 12 V puede mostrar grandes transistores de tensión cuando se la conecta y desconecta a la red. Si esta posibilidad existe, se recomienda emplear un circuito limitador de tensión.

CARACTERISTICAS ELECTRICAS (T_A = 0 °C a 70 °C, V_{CC} = +5 V ± 5 %, salvo que se indique otra cosa).

Parámetro	Mín.	Tip.	Máx.	Unidad	Comentario
CARACTERISTICAS EN C.C.					
NIVELES DE TENSION DE ENTRADA					
Nivel bajo, V_{IL}			0,8	V	
Nivel alto, V_{IH}	2,0			V	Excluyendo las entradas del XTAL
NIVELES DE TENSION DE SALIDA					
Nivel bajo, V_{OL}			0,4	V	I_{OL} = 1,6 mA, para $f_X/4$, $f_O/16$
			0,4	V	I_{OL} = 3,2 mA, para f_O, f_R, f_T
			0,4	V	I_{OL} = 0,8 mA, para f_X
Nivel alto, V_{OH}	3,5			V	I_{OH} = −100 μA; para f_X, I_{OH} = −50 μA
CORRIENTE DE ENTRADA					
Nivel bajo, I_{IL}			−0,1	mA	V_{IN} = GND, excluyendo entradas XTAL
CAPACITANCIA DE ENTRADA					
Todas las entradas, C_{IN}		5	10	pF	V_{IN} = GND, excluyendo entradas XTAL
CARGA EN LA ENTRADA EXTERNA		8	10		Carga equivalente a la serie 7400
CORRIENTE DE ALIMENTACION			50	mA	
I_{CC}					
CARACTERISTICAS EN C.A.					T_A = +25 °C
FRECUENCIA DEL RELOJ, f_{IN}	0,01		7,0	MHz	XTAL/EXT, 50 %. Ciclo de trabajo ± 5 %
					COM 8046, COM 8126, COM 8146
	0,01		5,1	MHz	XTAL/EXT, 50 %. Ciclo de trabajo ± 5 %
					COM 8116, COM 8136
ANCHURA DEL IMPULSO DE REFERENCIA, t_{PW}	150		DC	ns	
TIEMPO DE ESTABLECIMIENTO A LA ENTRADA t_{DS}	200			ns	
TIEMPO DE RETENCION A LA ENTRADA t_{DH}	50			ns	
RETARDO ENTRE IMPULSO DE REFERENCIA Y NUEVA FRECUENCIA			3,5	μs	@ f_X = 5,0 MHz







Para reprogramación de las ROM, SMC dispone de un programa de computadora a disposición del cliente, quien solo necesita suministrar la frecuencia de entrada y las frecuencias de salida deseadas. La programación de la ROM se produce automáticamente.

Especificaciones del cristal

El usuario debe especificar su terminación (terminales, hilos, otros)
Prefer.: HC-18/U o HC-25/U
Frecuencia: 5,0688 MHz. Corte AT
Margen de temperatura de 0 a 70°C
Resistencia serie 50Ω
Resonancia serie
Tolerancia global ±0,1 % o como se requiera

Lista parcial (Fabricantes de cristales)

Northern Engineering Laboratories
357 Beloit Street
Burlington, Wisconsin 53105
(414) 763-3591

Bulova Frequency Control Products
61-20 Woodside Avenue
Woodside, New York 11377
(212) 335-6000

CTS Knights Inc.
101 East Church Street
Sandwich, Illinois 60548
(815) 786-8411

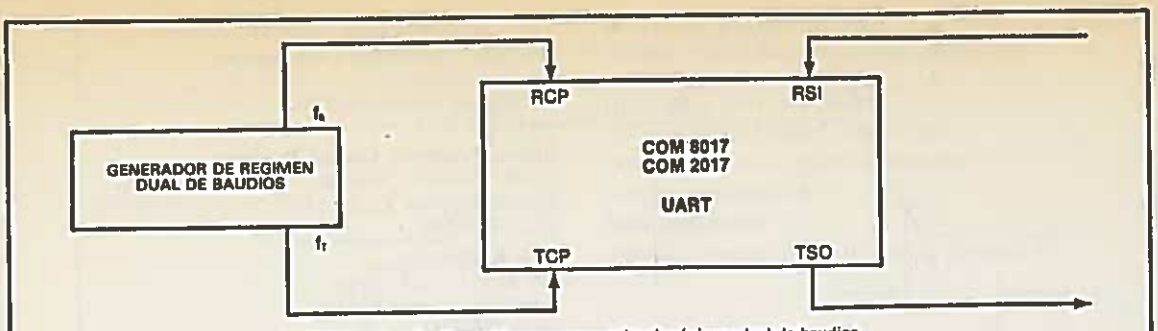
Crystek Crystals Corporation
1000 Crystal Drive
Fort Myers. Florida 33901
(813) 936-2109

COM 8046
COM 8046T

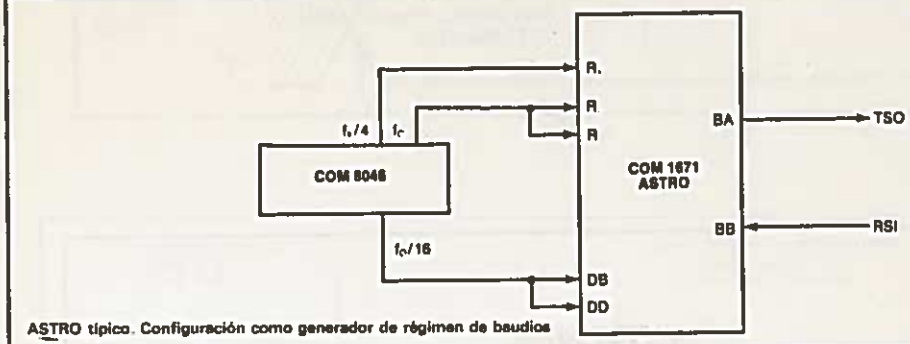
Tabla 2

FRECUENCIA DE REFERENCIA = 5,068800 MHz

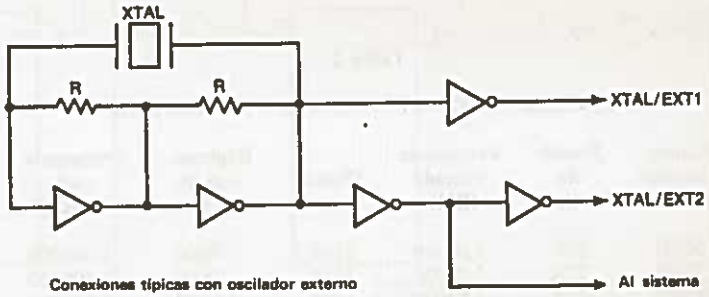
Selección del divisor	Régimen de baudios deseado	Factor de reloj	Frecuencia deseada (KHz)	Divisor	Régimen real de baudios	Frecuencia real (KHz)	Desviación %
00000	50,00	32X	1,60000	3168	50,00	1,600000	0,0000
00001	75,00	32X	2,40000	2112	75,00	2,400000	0,0000
00010	110,00	32X	3,52000	1440	110,00	3,520000	0,0000
00011	134,50	32X	4,30400	1177	134,58	4,306542	0,0591
00100	150,00	32X	4,80000	1056	150,00	4,800000	0,0000
00101	200,00	32X	6,40000	792	200,00	6,400000	0,0000
00110	300,00	32X	9,60000	528	300,00	9,600000	0,0000
00111	600,00	32X	19,20000	264	600,00	19,200000	0,0000
01000	1200,00	32X	38,40000	132	1200,00	38,400000	0,0000
01001	1800,00	32X	57,60000	88	1800,00	57,600000	0,0000
01010	2400,00	32X	76,80000	66	2400,00	76,800000	0,0000
01011	3600,00	32X	115,20000	44	3600,00	115,200000	0,0000
01100	4800,00	32X	153,60000	33	4800,00	153,600000	0,0000
01101	7200,00	32X	230,40000	22	7200,00	230,400000	0,0000
01110	9600,00	32X	307,20000	16	9900,00	316,800000	3,1250
01111	19 200,00	32X	614,40000	8	19 800,00	633,600000	3,1250
10000	50,00	16X	0,80000	6 336	50,00	0,800000	0,0000
10001	75,00	16X	1,20000	4224	75,00	1,200000	0,0000
10010	110,00	16X	1,76000	2880	110,00	1,760000	0,0000
10011	134,50	16X	2,15200	2355	134,52	2,152357	0,0166
10100	150,00	16X	2,40000	2112	150,00	2,400000	0,0000
10101	300,00	16X	4,80000	1056	300,00	4,800000	0,0000
10110	600,00	16X	9,60000	528	600,00	9,600000	0,0000
10111	1200,00	16X	19,20000	264	1200,00	19,200000	0,0000
11000	1800,00	16X	28,80000	176	1800,00	28,800000	0,0000
11001	2000,00	16X	32,00000	158	2005,06	32,081013	0,2532
11010	2400,00	16X	38,40000	132	2400,00	38,400000	0,0000
11011	3600,00	16X	57,60000	88	3600,00	57,600000	0,0000
11100	4800,00	16X	76,80000	66	4800,00	76,800000	0,0000
11101	7200,00	16X	115,20000	44	7200,00	115,200000	0,0000
11110	9600,00	16X	153,60000	33	9600,00	153,600000	0,0000
11111	19 200,00	16X	307,20000	16	19 800,00	316,800000	3,1250



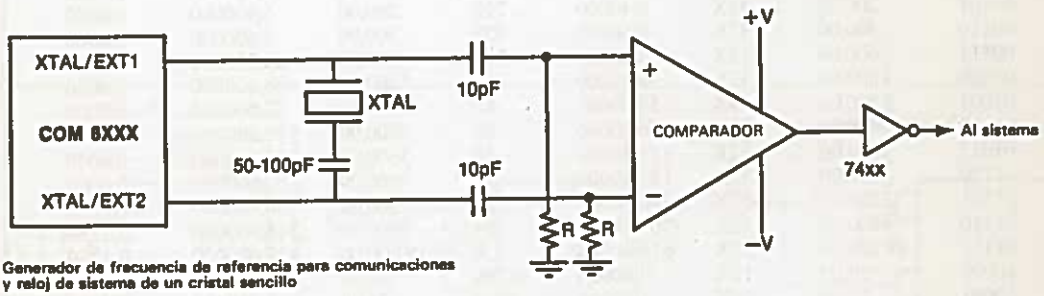
UART típico. Configuración como generador de régimen dual de baudios.
Dúplex completo. Velocidad partida



ASTRO típico. Configuración como generador de régimen de baudios



Conexiones típicas con oscilador externo

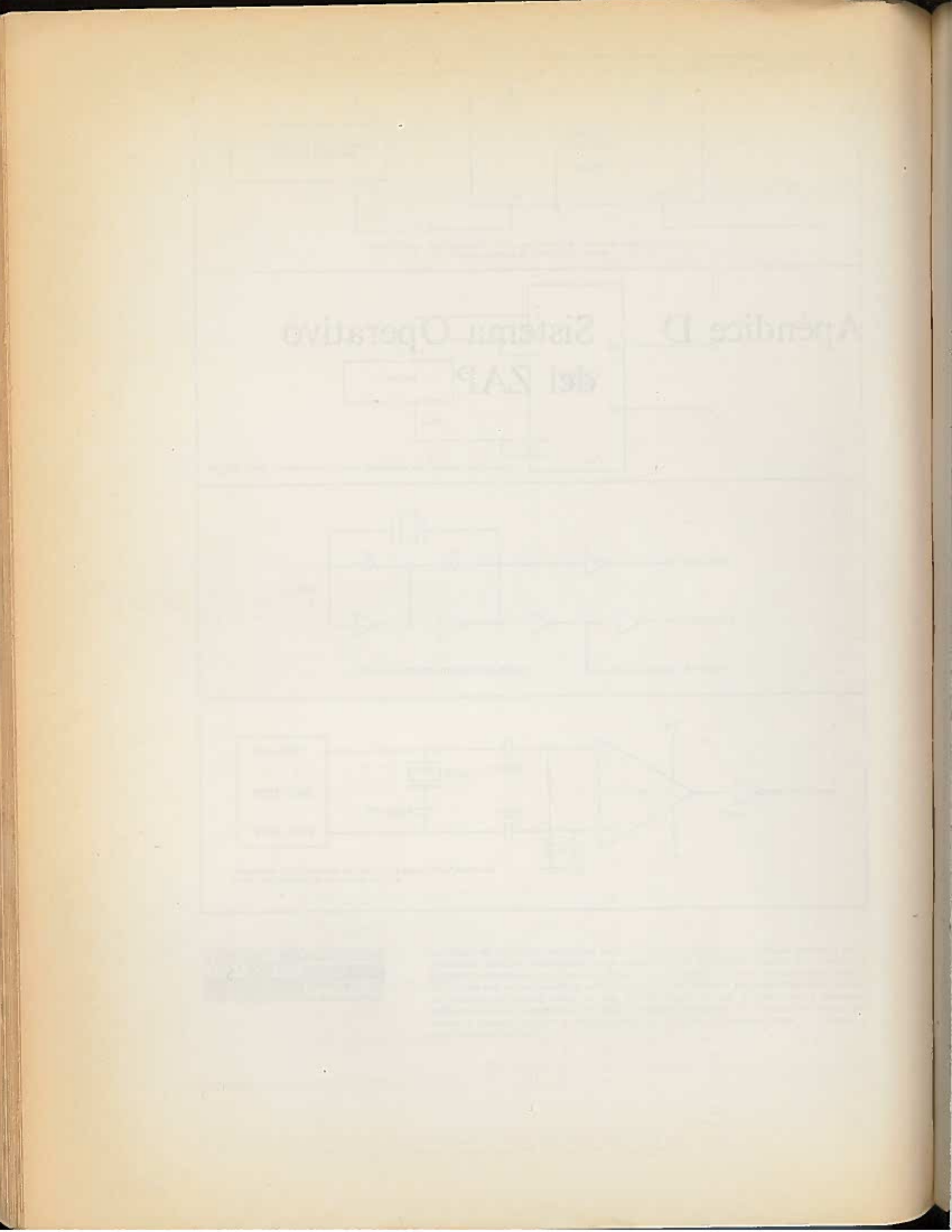


Generador de frecuencia de referencia para comunicaciones y reloj de sistema de un cristal sencillo

STANDARD MICROSYSTEMS CORPORATION

Los diagramas de circuitos que utilizan productos SMC se incluyen como medios de ilustración de las aplicaciones típicas de semiconductores; en consecuencia, no es necesario presentar una información completa y suficiente para las necesidades de diseño. La información ha sido comprobada cuidadosamente y se cree que es completamente fiable. Sin embargo, no se asume responsabilidad alguna por las inexactitudes que pudiera haber. Es más, tal información no cede al comprador el dispositivo semiconductor descrito permiso alguno sobre los derechos de patente de SMC u otros. SMC se reserva el derecho a introducir cambios en cualquier momento con el fin de mejorar el diseño y suministrar el mejor producto posible.

Apéndice D Sistema Operativo del ZAP



Apéndice D

Sistema Operativo del ZAP

```
FILE 3000 7323
READY
ASSM

7324                0100 *
7324                0110 *
7324                0120 *LAS SIGUIENTES EQUIVALENCIAS SE USAN
7324                0130 *COMO CONSTANTES DEL SISTEMA OPERATIVO
7324                0140 *
7324                0150 ZERO     EQU   0
7324                0160 ONE      EQU   1
7324                0170 TWO      EQU   2
7324                0180 THREE    EQU   3
7324                0190 FOUR     EQU   4
7324                0200 FIVE     EQU   5
7324                0210 EIGHT    EQU   8D
7324                0220 ADDIS1   EQU   5          *MSDS PORT PARA REPRESENTACION DE DIRECCIONES
7324                0230 ADDIS2   EQU   6          *LSDS PORT PARA REPRESENTACION DE DIRECCIONES
7324                0240 DATDIS   EQU   7          *PORT PARA REPRESENTACION DE DATOS
7324                0250 EXECC    EQU   16D        *TECLA DE EJECUCION
7324                0260 NEXTC    EQU   32D        *TECLA SIGUIENTE
7324                0270 UARTIO   EQU   2          *PORT DE ENTRADA/SALIDA PARA UART
7324                0280 UARTST   EQU   3          *PORT DE ESTADO PARA UART
7324                0290 KEYPT    EQU   0          *PORT DE ENTRADA PARA TECLADO
7324                0300 *
7324                0310 *
0000                0320          ST    0
0000                0330 *
0000                0340 *
0000                0350 *COLD ESTABLECE EL PUNTERO DE PILA PARA EL SISTEMA OPERATIVO
0000                0360 *E INTRODUCE EL MODULO DE RECONOCIMIENTO DE ORDENES
0000                0370 *
0000                0380 *
0000 31 C4 07       0390 COLD     LD    SP,SPSTRT    *INICIALIZA EL PUNTERO DE PILA
0003 C3 40 00       0400          JP    WARM01
0006                0410          DS    2
0008 C3 47 00       0420 WARM     JP    WARM1        *RST 1 OR WARM START
000B                0430          DS    5
0010 C3 C5 07       0440 RST2E    JP    RST2V        *TRANSFERENCIA RST 2
0013                0450          DS    5
0018 C3 C8 07       0460 RST3E    JP    RST3V        *TRANSFERENCIA RST 3
001B                0470          DS    5
0020 C3 CB 07       0480 RST4E    JP    RST4V        *TRANSFERENCIA RST 4
0023                0490          DS    5
0028 C3 CE 07       0500 RST5E    JP    RST5V        *TRANSFERENCIA RST 5
002B                0510          DS    5
0030 C3 D1 07       0520 RST6E    JP    RST6V        *TRANSFERENCIA RST 6
0033                0530          DS    5
0038 C3 D4 07       0540 RST7E    JP    RST7V        *TRANSFERENCIA RST 7
003B                0550          DS    5
0040 ED 73 DB 07    0551 WARM01   LD    (SPLSAV),SP
0044 C3 89 00       0552          JP    WARM2        *RRA A RECONOCIMIENTO DE ORDENES
0047                0560 *
0047                0570 *
0047                0580 *
0047                0590 *EL ARRANQUE EN CALIENTE CONSERVA LOS REGISTROS DEL USUARIO E
0047                0600 *INTRODUCE EL MODO DE RECONOCIMIENTO DE ORDENES CON
```

```
0047                0610 *FS REPRESENTADO EN LAS POSICIONES DE DATOS Y DIRECCIONES
0047                0620 *
0047 32 E3 07       0630 WARM1  LD    (ASAV),A       *CONSERVA EL A DEL USUARIO
004A E1             0640        POP   HL            *OBTENCION A PARTIR DE LA PILA EL PC PARA USUARIO
004B 22 DD 07       0650        LD    (PCLSAV),HL    *GUARDA EL PC PARA USUARIO EN ZONA DE
                                                      CONSERVACION
004E F5             0660        PUSH  AF
004F E1             0670        POP   HL            *OBTENCION INDICADORES DE USUARIO
0050 22 E7 07       0680        LD    (FSAV),HL      *GUARDA INDICADORES DE USUARIO
0053 DD 22 D7 07    0690        LD    (IXLSAV),IX    *GUARDA IX DE USUARIO
0057 FD 22 D9 07    0700        LD    (IYLSAV),IY    *GUARDA IY DE USUARIO
005B ED 73 DB 07    0710        LD    (SPLSAV),SP    *GUARDA SP DE USUARIO
005F ED 57          0720        LD    A,I            *GUARDA I DE USUARIO
0061 32 DF 07       0730        LD    (ISAV),A
0064 ED 5F          0740        LD    A,R            *GUARDA R DE USUARIO
0066 32 E0 07       0750        LD    (RSAV),A
0069 21 E4 07       0760        LD    HL,BSAV
006C 70             0770        LD    (HL),B         *GUARDA B DE USUARIO
006D 23             0780        INC   HL
006E 71             0790        LD    (HL),C         *GUARDA C DE USUARIO
006F 23             0800        INC   HL
0070 72             0810        LD    (HL),D         *GUARDA D DE USUARIO
0071 23             0820        INC   HL
0072 73             0830        LD    (HL),E         *GUARDA E DE USUARIO
0073 08             0840        EX    AF,AF          *GUARDA REGISTROS ALTERNOS
0074 F5             0850        PUSH  AF
0075 32 EB 07       0860        LD    (AASAV),A      *GUARDA PARADA EN A
0078 22 E9 07       0870        LD    (ALSAV),HL     *GUARDA PARADA EN HYL
007B E1             0880        POP   HL
007C 22 EF 07       0890        LD    (AFSAV),HL     *GUARDA INDICADORES DE PARADA
007F 21 EC 07       0900        LD    HL,ABSAV
0082 70             0910        LD    (HL),B         *GUARDA PARADA EN B
0083 23             0920        INC   HL
0084 71             0930        LD    (HL),C         *GUARDA PARADA EN C
0085 23             0940        INC   HL
0086 72             0950        LD    (HL),D         *GUARDA PARADA EN D
0087 23             0960        INC   HL
0088 73             0970        LD    (HL),E         *GUARDA PARADA EN E
0089                0980 *
0089                0990 *
0089                1000 *MODULO PARA RECONOCIMIENTO DE ORDENES
0089                1010 *
0089 CD F1 00       1020 WARM2  CALL  CLDIS          *BORRA LA PRESENTACION VISUAL
008C 3E FF          1030        LD    A,255D         *REPRESENTA FFFF FF
008E D3 05          1040        OUT   ADDIS1
0090 D3 06          1050        OUT   ADDIS2
0092 D3 07          1060        OUT   DATDIS
0094 CD 03 01       1070        CALL  KEYIN          *OBTENCION DE UN CARACTER DE ENTRADA
0097 06 40          1080        LD    B,MEM
0099 B8             1090        CP    B
009A CA F1 01       1100        JP    Z,MEMORY       *SALTO SI HAY SOLICITUD DE MEMORIA
009D 04             1110        INC   B
009E B8             1120        CP    B
009F CA 4B 02       1130        JP    Z,REGIST       *SALTO SI HAY SOLICITUD DE REGISTRO
00A2 04             1140        INC   B
00A3 B8             1150        CP    B
00A4 CA 10 03       1160        JP    Z,GOREQ
00A7 C3 89 00       1170        JP    WARM2
00AA                1180 *
00AA                1190 MEM    EQU   64D            *CLAVE DE MEMORIA
00AA                1200 *
00AA                1210 *
00AA                1220 *RESTART REINICIALIZA LOS REGISTROS DE USUARIO
00AA                1230 *Y DEVUELVE EL CONTROL A LA DIRECCION
00AA                1240 *ESPECIFICADA EN LA POSICION PARA GUARDAR EL PC
00AA                1250 *ZONA DE CONSERVACION DE REGISTROS
00AA                1260 *
00AA 3A EC 07       1270 RESTRT LD    A,(ABSAV)      *RESTAURAR LOS REGISTROS ALT
00AA 47             1280        LD    B,A
00AE 3A ED 07       1290        LD    A,(ACSAV)
00B1 4F             1300        LD    C,A
00B2 3A EE 07       1310        LD    A,(ADSAV)
00B5 57             1320        LD    D,A
00B6 3A EF 07       1330        LD    A,(AESAV)
00B9 5F             1340        LD    E,A
00BA 3A F0 07       1350        LD    A,(AFSAV)
00BD 6F             1360        LD    L,A
00BE E5             1370        PUSH  HL
```

```
00BF F1              1380          POP   AF
00C0 3A EB 07        1390          LD    A,(AASAV)
00C3 2A E9 07        1400          LD    HL,(ALSAV)
00C6 D9              1410          EXX
00C7 FD 2A D9 07     1420          LD    IY,(IYLSAV)    *RESTAURAR IY
00CB DD 2A D7 07     1430          LD    IX,(IXLSAV)    *RESTAURAR IX
00CF 21 DF 07        1440          LD    HL,ISAV
00D2 7E              1450          LD    A,(HL)
00D3 ED 47           1460          LD    I,A
00D5 23              1470          INC   HL
00D6 7E              1480          LD    A,(HL)
00D7 ED 4F           1490          LD    R,A
00D9 21 E3 07        1500          LD    HL,ASAV
00DC 7E              1510          LD    A,(HL)         *RESTAURAR A
00DD 23              1520          INC   HL
00DE 46              1530          LD    B,(HL)         *RESTAURAR B
00DF 23              1540          INC   HL
00E0 4E              1550          LD    C,(HL)         *RESTAURAR C
00E1 23              1560          INC   HL
00E2 56              1570          LD    D,(HL)         *RESTAURAR D
00E3 23              1580          INC   HL
00E4 5E              1590          LD    E,(HL)         *RESTAURAR E
00E5 ED 7B DB 07     1600          LD    SP,(SPLSAV)    *RESTAURAR EL PUNTERO DE PILA
00E9 2A DD 07        1610          LD    HL,(PCLSAV)    *REEMPLAZAR PC EN LA PILA
00EC E5              1620          PUSH  HL
00ED 2A E1 07        1630          LD    HL,(LSAV)      *RESTAURAR HYL
00F0 C9              1640          RET                  *DEVOLVER AL USUARIO
00F1                 1650 *
00F1                 1660 **
00F1                 1670 *
00F1                 1680 *CLDIS BORRA LAS PRESENTACIONES DE DATOS Y DIRECCIONES
00F1                 1690 *ACTIVA LA MEMORIA INTERMEDIA DEL TECLADO=0 Y BORRA LOS
00F1                 1700 *INDICADORES DEL TECLADO
00F1                 1710 *
00F1 3E 00           1720 CLDIS    LD    A,ZERO
00F3 32 F1 07        1730          LD    (KFLAGS),A     *BORRA INDICADORES
00F6 32 F2 07        1740          LD    (KDATA1),A     *BORRA MEMORIA INTERMEDIA
00F9 32 F3 07        1750          LD    (KDATA2),A
00FC D3 07           1760          OUT   DATDIS         *BORRA LA PRESENTACION DEL CAMPO DE DATOS
00FE D3 05           1770          OUT   ADDIS1         *BORRA LA PRESENTACION DEL CAMPO DE DIRECCIONES
0100 D3 06           1780          OUT   ADDIS2
0102 C9              1790          RET
0103                 1800 *
0103                 1810 *
0103                 1820 *KEYIN ESPERA INTERRUPCIONES DEL TECLADO
0103                 1830 *TRAS DETECTAR DATOS EN EL PORT DE ENTRADA A TRAVES DEL
0103                 1840 *BIT (7) DE REFERENCIA, QUE CUANDO ESTA A «UNO» LOS DATOS
0103                 1850 *SON DE ENTRADA, EL BIT DE REFERENCIA SE BORRA Y EL CARACTER
0103                 1860 *DE ENTRADA SE DEVUELVE AL USUARIO EN A
0103                 1870 *
0103                 1880 *
0103 DB 00           1890 KEYIN    IN    KEYPT          *INTRODUCIR DATOS
0105 CB 7F           1900          BIT   7,A
0107 CA 03 01        1910          JP    Z,KEYIN        *ENTRAR EN UN BUCLE SI NO HAY DATOS
010A 32 F4 07        1911          LD    (TEM),A        *GUARDAR EL CARACTER
010D DB 00           1912 KEYIN1   IN    KEYPT
010F CB 7F           1913          BIT   7,A
0111 C2 0D 01        1914          JP    NZ,KEYIN1      *SALTAR SI EL IMPULSO DE REFERENCIA ESTA PRESENTE
0114 3A F4 07        1915          LD    A,(TEMP)
0117 CB BF           1920          RES   7,A            *BORRAR EL IMPULSO DE REFERENCIA
0119 C9              1930          RET
011A                 1940 *
011A                 1950 *
011A                 1960 *KFLG02 PONE A «1» LOS INDICADORES DE TECLADO SIGUIENTE (0) Y DE «NO
011A                 1970 *DATOS» (2)
011A                 1980 *
011A 21 F1 07        1990 KFLG02   LD    HL,KFLAGS
011D CB C6           2000          SET   0,(HL)         *PONE A «1» EL INDICADOR SIGUIENTE
011F CB D6           2010          SET   2,(HL)
0121 E1              2020          POP   HL             *BORRA RETORNO
0122 C9              2030          RET
0123                 2040 *
0123                 2050 *
0123                 2060 *KFLGO PONE A «1» EL INDICADOR DE TECLADO PROXIMO
0123                 2070 *
0123                 2080 *
0123 21 F1 07        2090 KFLGO    LD    HL,KFLAGS
0126 CB C6           2100          SET   0,(HL)         *PONE A «1» EL INDICADOR SIGUIENTE
```

```
0128 E1          2110          POP   HL          *BORRA RETORNO
0129 C9          2120          RET
012A             2130 *
012A             2140 *
012A             2150 *KFLG12 PONE A «1» LOS INDICADORES DE TECLADO EJECUCION (1)
012A             2160 *Y «NO DATOS» (2)
012A             2170 *
012A 21 F1 07    2180 KFLG12   LD    HL,KLAGS
012D CB CE       2190          SET   1,(HL)
012F CB D6       2200          SET   2,(HL)
0131 E1          2210          POP   HL          *BORRAR RETORNO
0132 C9          2220          RET
0133             2230 *
0133             2240 *
0133             2250 *KFLG1 PONE A «1» EL INDICADOR DE TECLADO DE EJECUCION (1)
0133             2260 *
0133             2270 *
0133 21 F1 07    2280 KFLG1    LD    HL,KFLAGS
0136 CB CE       2290          SET   1,(HL)      *PONE A «1» EL INDICADOR DE EJECUCION
0138 E1          2300          POP   HL          *BORRAR RETORNO
0139 C9          2310          RET
013A             2320 *
013A             2330 *
013A             2340 *
013A             2350 *ONECAR INTRODUCE UN CARACTER SEGUIDO POR UN NEX O EXEC
013A             2360 *DEL TECLADO, LO VALIDA Y LO DEVUELVE
013A             2370 *AL USUARIO EN KDATA2
013A             2380 *
013A             2390 *
013A CD F1 00    2400 ONECAR   CALL  CLDIS       *BORRA LA PRESENTACION, LA MEMORIA INTERMEDIA Y
                                                   LOS INDICADORES
013D CD 03 01    2410          CALL  KEYIN       *OBTENCION DE UN CARACTER
0140 D3 07       2420          OUT   DATDIS      *REPRESENTA EL CARACTER
0142 CD 6D 01    2430          CALL  CARCK1      *COMPRUEBA EL CARACTER
0145 CB 77       2440          BIT   6,A
0147 C2 51 01    2450          JP    NZ,ONECA1   *SALTO, SI HAY DESPLAZAMIENTO
014A D6 10       2460          SUB   16D         *CARACTER=0-F
014C F2 3A 01    2470          JP    P,ONECAR    *SALTO SI NO 0-F
014F C6 10       2480          ADD   16D
0151 32 F3 07    2490 ONECA1   LD    (KDATA2),A  *GUARDE EL CARACTER
0154 CD 03 01    2500          CALL  KEYIN       *OBTENCION DEL SIGUIENTE CARACTER
0157 CD 6A 01    2510          CALL  CARCK2
015A C3 51 01    2520          JP    ONECA1      *PASA OTRA VEZ A DO SIN EXEC NI NEXT
015D             2530 *
015D             2540 *
015D             2550 *CARCK1 COMPRUEBA UN NEXT O EXEC EN UN CARACTER
015D             2560 *INICIAL. SI ES NEXT, LA RUTINA VUELVE ADONDE LA LLAMAN MEDIANTE
015D             2570 *VIA KFLGO2. SI ES EXEC, LA RUTINA VUELVE ADONDE LA LLAMAN MEDIANTE
015D             2580 *VIA KFLG12
015D             2590 *
015D             2600 *
015D 06 20       2610 CARCK1   LD    B,NEXTC     *COMPRUEBA SI HAY NEXT
015F B8          2620          CP    B
0160 CA 1A 01    2630          JP    Z,KFLGO2    *SALTO SI HAY NEXT
0163 06 10       2640          LD    B,EXECC     *COMPRUEBA SI HAY EXEC
0165 B8          2650          CP    B
0166 CA 2A 01    2660          JP    Z,KFLG12    *SALTO, SI HAY EXEC
0169 C9          2670          RET               *SI NO, RETORNA
016A             2680 *
016A             2690 *
016A             2700 *CARCK2 COMPRUEBA SI HAY NEXT O EXEC, PONE A «1» EL INDICADOR
016A             2710 *ADECUADO MEDIANTE KFLGO O KFLG1 Y RETORNA AL USUARIO
016A             2720 *SI NO HAY NEXT O EXEC, LA RUTINA VUELVE AL
016A             2730 *ORIGINADOR DE LA LLAMADA
016A             2740 *
016A             2750 *
016A 06 20       2760 CARCK2   LD    B,NEXTC     *COMPRUEBA SI HAY NEXT
016C B8          2770          CP    B
016D CA 23 01    2780          JP    Z,KFLGO     *SALTO SI HAY NEXT
0170 06 10       2790          LD    B,EXECC     *COMPRUEBA SI HAY EXEC
0172 B8          2800          CP    B
0173 CA 33 01    2810          JP    Z,KFLG1
0176 C9          2820          RET
0177             2830 *
0177             2840 *
0177             2850 *TWOCAR INTRODUCE DOS CARACTERES DEL TECLADO
0177             2860 *SEGUIDOS POR UN NEXT O EXEC Y DEVUELVE AL
0177             2870 *USUARIO EN KDATA2
```

```
0177                2880 *
0177                2890 *
0177 CD A0 01       2900 TWOCAR  CALL  CLDAT      *BORRA MEMORIA INTERMEDIA, INDICADORES Y
                                                  REPRESENTACION
017A CD 03 01       2910         CALL  KEYIN      *OBTENCION DE UN CARACTER
017D CD 5D 01       2930         CALL  CARCK1     *COMPRUEBA SI HAY NEXT O EXEC
0180 D6 10          2940 TWOCA1  SUB   16D        *CARACTER=0-F
0182 F2 77 01       2950         JP    P,TWOCAR   *SALTO SI NO 0-F
0185 C6 10          2960         ADD   16D
0187 21 F3 07       2970         LD    HL,KDATA2
018A 46             2980         LD    B,(HL)     *OBTENCION DE DATOS ANTIGUOS
018B CB 00          2990         RLC   B
018D CB 00          3000         RLC   B
018F CB 00          3010         RLC   B
0191 CB 00          3020         RLC   B
0193 80             3030         ADD   A,B        *A=ANTIGUO Y NUEVO
0194 D3 07          3031         OUT   DATDIS     *VISUALIZACION DE LA ENTRADA
0196 77             3040         LD    (HL),A     *GUARDA DATO ANTIGUO
0197 CD 03 01       3050         CALL  KEYIN      *OBTENCION DEL SIGUIENTE CARACTER
019A CD 6A 01       3060         CALL  CARCK2     *COMPRUEBA SI HAY TERMINACION
019D C3 80 01       3070         JP    TWOCA1     *SALTO SI NO HAY TERMINACION
01A0                3080 *
01A0                3090 *
01A0                3100 *CLDAT BORRA LA MEMORIA INTERMEDIA DE ENTRADA Y LA REPRESENTACION DE DATOS
01A0                3110 *
01A0                3120 *
01A0 3E 00          3130 CLDAT   LD    A,ZERO
01A2 32 F1 07       3140         LD    (KFLAGS),A *BORRA INDICADORES
01A5 32 F3 07       3150         LD    (KDATA2),A *BORRA MEMORIA INTERMEDIA
01A8 32 F2 07       3160         LD    (KDATA1),A
01AB C9             3180         RET
01AC 3E 00          3181 CLADD   LD    A,ZERO     *BORRA REPRESENTACION DE DIRECCIONES
01AE D3 05          3182         OUT   ADDIS1
01B0 D3 06          3183         OUT   ADDIS2
01B2 C9             3184         RET
01B3                3190 *
01B3                3200 *
01B3                3210 *FORCAR INTRODUCE CUATRO CARACTERES DEL TECLADO
01B3                3220 *SEGUIDOS POR UN NEXT O EXEC Y LOS DEVUELVE
01B3                3230 *AL USUARIO EN KDATA1 Y KDATA2
01B3                3240 *
01B3                3250 *
01B3 CD A0 01       3260 FORCAR  CALL  CLDAT      *BORRA INDICADORES Y MEMORIA INTERMEDIA
01B6 CD 03 01       3270         CALL  KEYIN      *OBTENCION DE UN CARACTER DE ENTRADA
01B9 CD 5D 01       3280         CALL  CARCK1     *COMPRUEBA SI HAY NEXT O EXEC
01BC D6 10          3290 FORCA1  SUB   16D        *CARACTER=0-F
01BE F2 B3 01       3300         JP    P,FORCAR   *SALTO SI NO 0-F
01C1 C6 10          3310         ADD   16D
01C3 32 F4 07       3320         LD    (TEMP),A   *GUARDA EL CARACTER
01C6 3A F2 07       3330         LD    A,(KDATA1) *A=MSD
01C9 21 F3 07       3340         LD    HL,KDATA2
01CC ED 67          3350         RRD              *AJUSTA DATO AL NUEVO CARACTER
01CE 07             3360         RLCA
01CF 07             3370         RLCA
01D0 07             3380         RLCA
01D1 07             3390         RLCA
01D2 E6 F0          3400         AND   240D       *RETIRA LA MASCARA AL IDGIT ANTIGUO
01D4 21 F4 07       3410         LD    HL,TEMP
01D7 86             3420         ADD   A,(HL)     *SUMA EL DIGITO NUEVO
01D8 2A F3 07       3430         LD    HL,(KDATA2) *GUARDA EL LSDS NUEVO
01DB 22 F2 07       3440         LD    (KDATA1),HL *GUARDA EL MSDS NUEVO
01DE 32 F3 07       3450         LD    (KDATA2),A *GUARDA EL LSDS NUEVO
01E1 D3 06          3460         OUT   ADDIS2     *REPRESENTA EL LSDS
01E3 3A F2 07       3470         LD    A,(KDATA1)
01E6 D3 05          3480         OUT   ADDIS1
01E8 CD 03 01       3490         CALL  KEYIN      *OBTENCION DEL CARACTER SIGUIENTE
01EB CD 6A 01       3500         CALL  CARCK2     *COMPRUEBA SI HAY NEXT O EXEC
01EE C3 BC 01       3510         JP    FORCA1     *SALTO SI HAY NEXT O EXEC
01F1                3520 *
01F1                3530 *
01F1                3540 *
01F1                3550 *
01F1                3560 *ENTRADAS Y DIRECCIONES DE MEMORIA DEL TECLADO SEGUIDAS
01F1                3570 *POR DATOS DEFINIDOS POR LA SECUENCIA
01F1                3580 *MEM (DIRECCION) NEXT, (DATO) NEXT... (DATO) EXEC
01F1                3590 *SI DATO ES PARA REPRESENTAR
01F1                3600 *MEM (DIRECCION) NEXT, (DATO) NEXT... (DATO) EXEC
01F1                3610 *EXEC DEVOLVERA EL CONTROL AL RECONOCIMIENTO DE ORDENES
```

```
01F1                3620 *
01F1                3630 *
01F1 3E 00          3640 MEMORY  LD    A,ZERO        *BORRA LA DIRECCION DE BASE DE MEMORIA
01F3 32 F6 07       3650         LD    (MBASE1),A
01F6 32 F7 07       3660         LD    (MBASE2),A
01F9 CD AC 01       3661         CALL  CLADD
01FC CD B3 01       3670         CALL  FORCAR        *OBTENER DIRECCION BASE
01FF 3A F1 07       3680         LD    A,(KFLAGS)
0202 CB 4F          3690         BIT   1,A
0204 C2 89 00       3700         JP    NZ,WARM2      *SALTO SI EL INDICADOR DE EXEC ESTA PUESTO A «UNO»
0207 3A F2 07       3710         LD    A,(KDATA1)    *GUARDA DIRECCION DE MEMORIA
020A 32 F7 07       3720         LD    (MBASE2),A
020D 3A F3 07       3730         LD    A,(KDATA2)
0210 32 F6 07       3740         LD    (MBASE1),A
0213 2A F6 07       3750         LD    HL,(MBASE1)   *ACTIVA LA DIRECCION DE BASE DE MEMORIA
0216 7E             3760 MEM1    LD    A,(HL)        *OBTENCION DE DATO DE LA MEMORIA
0217 D3 07          3770         OUT   DATDIS        *REPRESENTA EL DATO DE MEMORIA
0219 CD 77 01       3780         CALL  TWOCAR        *OBTENCION DE DATO NUEVO
021C 3A F1 07       3790         LD    A,(KFLAGS)
021F CB 57          3800         BIT   2,A
0221 C2 43 02       3810         JP    NZ,NEM2       *SALTO SI NO HAY DATO
0224 2A F6 07       3820         LD    HL,(MBASE1)   *OBTENCION DE LA DIRECCION DE LA MEMORIA
0227 3A F3 07       3830         LD    A,(KDATA2)    *OBTENCION DE DATO NUEVO
022A 77             3840         LD    (HL),A        *SUSTITUYE AL DATO ANTIGUO
022B 3A F1 07       3850         LD    A,(KFLAGS)
022E CB 4F          3860         BIT   1,A
0230 C2 89 00       3870         JP    NZ,WARM2      *SALTO SI EL INDICADOR DE EXEC ESTA PUESTO A «UNO»
0233 2A F6 07       3880 MEM12   LD    HL,(MBASE1)   *INC. DIRECCION BASE DE MEMORIA
0236 23             3890         INC   HL
0237 22 F6 07       3900         LD    (MBASE1),HL
023A 7D             3901         LD    A,L
023B D3 06          3902         OUT   ADDIS2
023D 7C             3903         LD    A,H
023E D3 05          3904         OUT   ADDIS1
0240 C3 16 02       3910         JP    MEM1
0243 CB 4F          3920 MEM2    BIT   1,A
0245 G2 89 00       3930         JP    NZ,WARm2      *SALTO SI EL INDICADOR DE EXEC ESTA PUESTO A «UNO»
0248 C3 33 02       3940         JP    MEM12
024B                3950 *
024B                3960 *
024B                3970 *
024B                3980 *
024B                3990 *REGIST INTRODUCE UN REGISTRO DEL TECLADO SEGUIDO POR
024B                4000 *UN DATO, TAL Y COMO SE LE DEFINE POR LA SECUENCIA
024B                4010 *REG (INIT REG) NEXT, (DATO) NEXT... (DATO) EXEC
024B                4020 *LA SECUENCIA DE REGISTROS ES IX, IY, SP, PC, I, R, H, L, A, B, C, D, E, F,
024B                4030 *AL, AH, AA, AB, AC, AD, AE, AF
024B                4040 *SI SOLO VA A SER REPRESENTADO EL DATO
024B                4050 *REG (INIT REG) NEXT, NEXT... EXEC
024B                4060 *EXEC DEVOLVERA EL CONTROL AL RECONOCIMIENTO DE ORDENES
024B                4070
024B                4080 *
024B CD 3A 01       4090 REGIST  CALL  ONECAR        *OBTENCION DEL CARACTER INICIAL
024E 3A F1 07       4100         LD    A,(KFLAGS)
0251 CB 57          4110         BIT   2,A
0253 C2 89 00       4120         JP    NZ,WARM2      *SALTO SI NO HAY PUESTO A «UNO» ALGUN INDICADOR DE
                                                       DATOS
0256 3A F3 07       4130         LD    A,(KDATA2)    *OBTENCION DEL REGISTRO BASE
0259 32 F5 07       4140 REGIO   LD    (TEMP2),A
025C CB 77          4141         BIT   6,A           *COMPROBAR SI HAY DESPLAZAMIENTO
025E C2 CC 02       4142         JP    NZ,REGISA     *SALTO SI LA CLAVE DE DESPLAZAMIENTO ESTA PUESTA A
                                                       «UNO»
0261 FE 06          4143         CP    6
0263 F2 6C 02       4144         JP    P,REGI1       *SALTO SI ES UN REGISTRO DE 8 BITS
0266 3D             4145         DEC   A
0267 3D             4146         DEC   A
0268 87             4147         ADD   A             *I=(I-2)*2
0269 C3 6E 02       4148         JP    REGI2
026C 3C             4149 REGI1   INC   A
026D 3C             4150         INC   A
026E 32 F8 07       4151 REGI2   LD    (REGINX),A    *GUARDA EL INDICE
0271 3A F5 07       4152         LD    A,(TEMP2)
0274 FE 10          4153         CP    10H
0276 FA 83 02       4154         JP    M,REGI2A
0279 CB 77          4155         BIT   6,A
027B C2 83 02       4157         JP    NZ,REGI2A     *SALTO SI EL BIT 6 ESTA PUESTO A «UNO»
027E 3E 48          4158         LD    A,48H
0280 32 F5 07       4159         LD    (TEMP2),A
```

```
0283 D3 07      4160 REGI2A  OUT   DATDIS      *SELECCION DEL REGISTRO DE REPRESENTACION
0285 3A F8 07   4210         LD    A,(REGINX)
0288 FE 08      4220         CP    EIGHT
028A FA D6 02   4230         JP    M,XYSP      *SALTO SI ES UN REGISTRO DE 16 BITS
028D 21 D7 07   4240         LD    HL,IXLSAV   *OBTENCION DE LA DIRECCION BASE
0290 4F         4250         LD    C,A
0291 06 00      4260         LD    B,ZERO
0293 09         4270         ADD   HL,BC

0294 22 F6 07   4280         LD    (MBASE1), HL *GUARDA REGISTRO, GUARDA DIRECCION
0297 7E         4290         LD    A,(HL)      *OBTENCION DE DATO DEL REGISTRO
0298 D3 06      4300         OUT   ADDIS2      *REPRESENTA EL DATO
029A 78         4310         LD    A,B
029B D3 05      4320         OUT   ADDIS1
029D CD 77 01   4330         CALL  TWOCAR      *OBTENCION DE DATO NUEVO
02A0 3A F1 07   4340         LD    A,(KFLAGS)
02A3 CB 57      4350         BIT   2,A
02A5 C2 B7 02   4360         JP    NZ,REGI3    *SALTO SI NO HAY DATOS
02A8 2A F6 07   4390         LD    HL,(MBASE1)
02AB 3A F2 07   4400         LD    A,(KDATA1)  *OBTENCION DE DATO NUEVO
02AE 77         4410         LD    (HL),A      *SUSTITUIR EL DATO ANTIGUO
02AF 3A F1 07   4411         LD    A,(KFLAGS)
02B2 CB 4F      4412         BIT   1,A
02B4 C2 89 00   4413         JP    NZ,WARM2    *SALTO SI EL INDICADOR DE EXEC ESTA PUESTO A «UNO»
02B7 3A F5 07   4420 REGI3   LD    A,(TEMP2)   *INCREMENTA INDICE
02BA 3C         4421         INC   A
02BB 32 F5 07   4422         LD    (TEMP2),A
02BE 3A F8 07   4423         LD    A,(REGINX)  *INCREMENTA INDICE
02C1 3C         4430         INC   A
02C2 FE 1A      4440         CP    1AH
02C4 FA 6E 02   4450         JP    M,REGI2     *SALTO SI EL INDICE .LT. 1A
02C7 3E 02      4460 REGIA   LD    A,TWO       *ACTIVAR EL INDICE INICIAL
02C9 C3 59 02   4470         JP    REGIO
02CC D6 48      4480 REGI8A  SUB   48H
02CE FA 4B 02   4490         JP    M,REGIST    *SALTO SI EL REGISTRO NO ES VALIDO
02D1 C6 12      4500         ADD   12H
02D3 C3 6E 02   4510         JP    REGI2
02D6 21 D7 07   4520 XYSP    LD    HL,IXLSAV
02D9 4F         4530         LD    C,A
02DA 06 00      4540         LD    B,ZERO
02DC 09         4550         ADD   HL,BC       *HL=DIRECCION DEL REGISTRO A GUARDAR

02DD 22 F6 07   4560         LD    (MBASE1),HL
02E0 7E         4570         LD    A,(HL)      *REPRESENTA DATOS DEL REGISTRO
02E1 D3 06      4580         OUT   ADDIS2
02E3 23         4590         INC   HL
02E4 7E         4600         LD    A,(HL)
02E5 D3 05      4610         OUT   ADDIS1
02E7 3A F8 07   4620         LD    A,(REGINX)
02EA 3C         4630         INC   A
02EB 32 F8 07   4640         LD    (REGINX),A
02EE CD B3 01   4650         CALL  FORCAR      *OBTENCION DE DATO NUEVO
02F1 3A F1 07   4660         LD    A,(KFLAGS)
02F4 CB 57      4670         BIT   2,A
02F6 C2 08 03   4680         JP    NZ,REGI6    *SALTO SI NO HAY DATOS
02F9 2A F6 07   4710         LD    HL,(MBASE1) *SUSTITUIR EL DATO VIEJO
02FC 3A F3 07   4720         LD    A,(KDATA2)
02FF 77         4730         LD    (HL),A
0300 3A F2 07   4740         LD    A,(KDATA1)
0303 23         4750         INC   HL
0304 77         4760         LD    (HL),A
0305 3A F1 07   4761         LD    A,(KFLAGS)
0308 CB 4F      4762 REGI6   BIT   1,A
030A C2 89 00   4763         JP    NZ,WARM2    *SALTO SI EL INDICADOR DE EXEC ESTA PUESTO A «UNO»
030D C3 B7 02   4770         JP    REGI3
0310            4780 *
0310            4790 *
0310            4800 *
0310            4810 *
0310            4820 *GO PONE A «CERO» LA DIRECCION DE RELANZAMIENTO DEL USUARIO
0310            4830 *ZONA DE GUARDAR REGISTROS Y SALE AL MODULO DE
0310            4840 *RELANZAMIENTO
0310            4850 *
0310            4860 *
0310 CD AC 01   4870 GOREQ   CALL  CLADD
0313 CD B3 01   4871         CALL  FORCAR      *OBTENCION DE LA DIRECCION PARA RELANZAMIENTO
0316 3A F1 07   4880         LD    A,(KFLAGS)
0319 CB 57      4890         BIT   2,A
```

```
031B C2 89 00     4900          JP    NZ,WARM2     *SI NO, SALEN DATOS
031E 3A F3 07     4910          LD    A,(KDATA2)   *GUARDA LA DIRECCION NUEVA
0321 32 DD 07     4920          LD    (PCLSAV),A
0324 3A F2 07     4930          LD    A,(KDATA1)
0327 32 DE 07     4940          LD    (PCHSAV),A
032A C3 AA 00     4950          JP    RESTRT
032D              4960 *
032D              4970 *
032D              4980 *
032D              4990 *UATST ES UNA RUTINA PARA COMPROBAR EL BUCLE UART
032D              5000 *UTILIZA UN BUCLE CON EL PORT DE
032D              5010 *SALIDA CONECTADA AL PORT DE ENTRADA
032D              5020 *SI SE DETECTA ALGUN ERROR; EL ERROR ES
032D              5030 *REPRESENTADO EN EL VISUALIZADOR DE DIRECCIONES Y
032D              5040 *EL CARACTER SE REPRESENTA EN EL DE DATOS
032D              5050 *EL CARACTER DE SALIDA SE REPRESENTA EN EL MSD
032D              5060 *DEL VISUALIZADOR DE DIRECCIONES
032D              5070 *
032D 06 00        5080 UATST    LD    B,ZERO       *
032F DB 03        5090          IN    UARTST       *OBTENCION DEL ESTADO
0331 CB 47        5100          BIT   0,A
0333 CA 53 03     5110          JP    Z,UAER1      *SALTO SI XMIT BUFFER NO ESTA VACIO
0336 78           5120 UATSTO   LD    A,B          *OBTENCION DEL CARACTER DE SALIDA
0337 D3 06        5130          OUT   ADDIS1
0339 D3 02        5140          OUT   UARTIO
033B DB 03        5150 UATST1   IN    UARTST
033D CB 4F        5160          BIT   1,A
033F CA 3B 03     5170          JP    Z,UATST1     *SALTO SI NO HAY DATOS DISPONIBLES
0342 E6 1C        5180          AND   1CH
0344 C2 53 03     5190          JP    NZ,UAER1     *SALTO SI HAY ERROR DE PARIDAD
0347 DB 02        5240          IN    UARTIO       *OBTENCION DE UN CARACTER DE ENTRADA
0349 D3 07        5250          OUT   DATDIS
034B B8           5260          CP    B
034C C2 5A 03     5270          JP    NZ,UAER2     *SALTO SI INPUT .NE. OUTPUT
034F 04           5280          INC   B
0350 C3 36 03     5290          JP    UATSTO
0353 D3 06        5300 UAER1    OUT   ADDIS2       *REPRESENTA EL ESTADO DEL UART
0355 DB 02        5310          IN    UARTIO       *OBTENCION DATO DE ENTRADA
0357 D3 07        5320          OUT   DATDIS
0359 76           5330          HALT
035A 3E 0F        5340 UAER2    LD    A,0FH
035C ED 79        5350          OUT   (ADDIS2),A
035E 76           5360          HALT
035F              5370 *
035F              5380 *
035F              5390 *CONTROLADOR PARA LA ENTRADA DEL TELETIPO
035F              5400 *INTRODUCE LOS DATOS EN LA MEMORIA INTERMEDIA ESPECIFICADA
035F              5410 *LA ENTRADA SE TERMINA CUANDO UN RETORNO DEL CARRO
035F              5420 *SE DETECTA O EL NUMERO ESPECIFICADO DE CARACTERES
035F              5430 *HA SIDO INTRODUCIDO POR EL DISPOSITIVO TRANSMISOR
035F              5440 *
035F 2A F9 07     5450 TTYINP   LD    HL,(TTYIBF)  *OBTENCION DE LA DIRECCION DE LA MEMORIA
                                                     INTERMEDIA
0362 3A FD 07     5460          LD    A,(TTYIC)    *OBTENCION DEL NUMERO DE CARACTERES
0365 47           5470          LD    B,A
0366 DB 03        5480 TTYIN1   IN    UARTST       *OBTENCION DEL ESTADO DEL UART
0368 CB 4F        5490          BIT   1,A
036A CA 66 03     5500          JP    Z,TTYIN1     *SALTO SI NO HAY DATOS
036D E6 1C        5510          AND   1CH
036F C2 9B 03     5520          JP    NZ,TTYERR    *SALTO SI HAY ERRORES DE PARIDAD
0372 DB 02        5570          IN    UARTIO       *OBTENCION DEL CARACTER DE ENTRADA
0374 77           5580          LD    (HL),A       *GUARDAR CARACTERES EN LA MEMORIA INTERMEDIA
                                                     PARA USUARIO
0375 FE 0D        5590          CP    A,0DH
0376 CA 91 03     5600          JP    Z,TTYIN2     *SALTO SI HAY RETORNO DEL CARRO
0379 3E 01        5610          LD    A,ONE        *ESTABLECE LA CUENTA DE CARACTERES DE SALIDA
037B 22 FB 07     5620 TTYIN3   LD    (TTYOBF),HL  *ESTABLECE DIRECCION DE MEMORIA INTERMEDIA DE
                                                     SALIDA
037E 32 FE 07     5630          LD    (TTYOC),A
0381 78           5631          LD    A,B
0382 32 F4 07     5632          LD    (TEMP),A
0385 CD 9E 03     5640          CALL  TTYOUT       *PASA AL CARACTER DE SALIDA
0388 3A F4 07     5641          LD    A,(TEMP)
038B 47           5642          LD    B,A
038C 05           5650          DEC   B
038D C8           5660          RET   Z            *RETORNO SI TODOS LOS CARACTERES ESTAN DENTRO
038E C3 66 03     5670          JP    TTYIN1
0391 21 9C 03     5680 TTYIN2   LD    HL,LF        *OBTENCION DE LA DIRECCION DEL AVANCE DE LINEAS
```

```
0394 3E 02          5690            LD      A,TWO
0396 06 01          5700            LD      B,ONE
0398 C3 7B 03       5710            JP      TTYIN3
039B C9             5720 TTYERR     RET                     *RETORNA CON EL CODIGO DE ERRORES A UN
039C 0D 0A          5730 LF         DB      0DH,0AH         *AVANCE DE LINEA/RETORNO DEL CARRO

039E                5740 *
039E                5750 *CONTROLADOR PARA LA SALIDA DEL TELETIPO
039E                5760 *TTYOUT SACA UN DATO DE LA MEMORIA INTERMEDIA
039E                5770 *PARA USUARIO ESPECIFICADA AL UART. EL NUMERO DE
039E                5780 *CARACTERES ESPECIFICADOS POR EL USUARIO SON SACADOS
039E                5790 *Y EL CONTROL DEVUELTO AL USUARIO
039E                5800 *
039E 2A FB 07       5810 TTYOUT     LD      HL,(TTYOBF)     *OBTENCION DE LA DIRECCION DE LA MEMORIA
                                                             INTERMEDIA
03A1 3A FE 07       5820            LD      A,(TTYOC)       *OBTENCION DEL NUMERO DE CARACTERES
03A4 47             5830            LD      B,A
03A5 0E 00          5840 TTYOU1     LD      C,ZERO
03A7 11 00 00       5850            LD      DE,ZERO
03AA DB 03          5860 TTYO1      IN      UARTST          *OBTENCION DEL ESTADO
03AC CB 47          5870            BIT     0,A
03AE CA BC 03       5880            JP      Z,TTYOU2        *SALTO SI LA MEMORIA INTERMEDIA NO ESTA VACIA
03B1 7E             5890            LD      A,(HL)          *OBTENCION DE UN CARACTER
03B2 D3 02          5900            OUT     UARTIO          *SACA UN CARACTER
03B4 05             5910            DEC     B
03B5 3E 00          5920            LD      A,ZERO
03B7 C8             5930            RET     Z               *RETORNO SI LA MEMORIA INTERMEDIA ESTA VACIA
03B8 23             5931            INC     HL
03B9 C3 A5 03       5940            JP      TTYOU1
03BC 13             5950 TTYOU2     INC     DE              RETARDO PARA PROBAR OTRA VEZ
03BD 7B             5960            LD      A,E
03BE FE 00          5970            CP      ZERO
03C0 C2 BC 03       5980            JP      NZ,TTYOU2
03C3 7A             5990            LD      A,D
03C4 FE 00          6000            CP      ZERO
03C6 C2 BC 03       6010            JP      NZ,TTYOU2
03C9 0C             6020            INC     C
03CA FE 05          6030            CP      FIVE
03CC C2 AA 03       6040            JP      NZ,TTYO1        *SALTO SI .LT. 5 PRUEBA
03CF 3E 01          6050            LD      A,ONE           *SINO, VOLVER A A=1
03D1 C9             6060            RET
03D2                6070 *
07C4                6080            ST      7C4H
07C4                6090 *
07C4                6100 *CONSTANTES DE LA PAGINA 2, ZONAS DE SALTO Y ZONAS
07C4                6110 *DE GUARDAR REGISTROS
07C4                6120 *
07C4                6130 SPSTRT     DB      0               *ZONA DE LA PILA
  00
07C5                6140 *
07C5                6150 *ZONA DE REINICIACION PARA EL USUARIO
07C5                6160 *
07C5                6170 RST2V      DS      3               *ZONA DE BIFURCACION PARA RST 2
07C8                6180 RST3V      DS      3               *ZONA DE BIFURCACION PARA RST 3
07CB                6190 RST4V      DS      3               *ZONA DE BIFURCACION PARA RST 4
07CE                6200 RST5V      DS      3               *ZONA DE BIFURCACION PARA RST 5
07D1                6210 RST6V      DS      3               *ZONA DE BIFURCACION PARA RST 6
07D4                6220 RST7V      DS      3               *ZONA DE BIFURCACION PARA RST 7
07D7                6230 *
07D7                6240 *ZONA DE GUARDAR REGISTROS
07D7                6250 *
07D7                6260 IXLSAV     DB      0
  00
07D8                6270 IXHSAV     DB      0
  00
07D9                6280 IYLSAV     DB      0
  00
07DA                6290 IYHSAV     DB      0
  00
07DB                6300 SPLSAV     DB      0
  00
07DC                6310 SPHSAV     DB      0
  00
07DD                6320 PCLSAV     DB      0
  00
07DE                6330 PCHSAV     DB      0
  00
07DF                6340 ISAV       DB      0
```

```
  00
07E0              6350 RSAV     DB    0
  00
07E1              6360 LSAV     DB    0
  00
07E2              6370 HSAV     DB    0
  00
07E3              6380 ASAV     DB    0
  00
07E4              6390 BSAV     DB    0
  00
07E5              6400 CSAV     DB    0
  00
07E6              6410 DSAV     DB    0
  00
07E7              6420 ESAV     DB    0
  00
07E8              6430 FSAV     DB    0
  00
07E9              6440 ALSAV    DB    0
  00
07EA              6450 AHSAV    DB    0
  00
07EB              6460 AASAV    DB    0
  00
07EC              6470 ABSAV    DB    0
  00
07ED              6480 ACSAV    DB    0
  00
07EE              6490 ADSAV    DB    0
  00
07EF              6500 AESAV    DB    0
  00
07F0              6510 AFSAV    DB    0
  00
07F1              6520 *
07F1              6530 *ZONA DE ALMACENAMIENTO DE DATOS
07F1              6540 *
07F1              6550 KFLAGS   DB    0          *INDICADORES DEL TECLADO
  00
07F2              6560 KDATA1   DB    0          *MEMORIA INTERMEDIA DE ENTRADA PARA EL TECLADO
  00
07F3              6570 KDATA2   DB    0
  00
07F4              6580 TEMP     DB    0
  00
07F5              6581 TEMP2    DB    0
  00
07F6              6590 MBASE1   DB    0          *DIRECCION DE MEMORIA BASE
  00
07F7              6600 MBASE2   DB    0
  00
07F8              6610 REGINX   DB    0          *REGISTRO INDICE
  00
07F9              6620 TTYIBF   DS    2          *DIRECCION DE MEMORIA INTERMEDIA DE ENTRADA
                                                  PARA TELETIPO
07FB              6630 TTYOBF   DS    2          *DIRECCION DE MEMORIA INTERMEDIA DE SALIDA PARA
                                                  TELETIPO
07FD              6640 TTYIC    DB    0          *CUENTA DE CARACTERES DE ENTRADA AL TELETIPO
  00
07FE              6650 TTYOC    DB    0          *CUENTA DE CARACTERES DE SALIDA AL TELETIPO
  00
07FF              6660 *
07FF              6670 *FIN


FILE  3000 7323
READY
```

Apéndice E

Especificaciones técnicas de la CPU del Z80

Apéndice E1

Especificaciones eléctricas

Regímenes máximos absolutos*

Temperatura bajo polarización	Alcance específico de la operación
Temperatura de almacenamiento	−65°C a +150°C
Tensión en cualquier terminal con respecto a tierra	−0,3 V a +7 V
Disipación de potencia	1,5 W

*COMENTARIO: Los excesos por encima de los valores indicados bajo los «regímenes máximos absolutos» pueden ocasionar daños permanentes al dispositivo. Esto es un régimen de exceso solamente y el funcionamiento del dispositivo no está implicado bajo estas u otras condiciones por encima de las indicadas en el apartado operativo de estas especificaciones. La exposición durante largos períodos a las condiciones de los regímenes máximos absolutos puede afectar a la fiabilidad del dispositivo.

NOTA: Para la CPU-Z80 todas las características permanecen idénticas para los modelos de tipo militar excepto la I_{CC}.

$$I_{CC} = 200 \text{ mA}$$

CARACTERISTICAS EN C.C. DE LA CPU-Z80
$T_A = 0$ °C a 70 °C, $V_{CC} = 5\text{ V} \pm 5\ \%$, salvo que se indique otra cosa.

Símbolo	Parámetro	Mín.	Típ.	Máx.	Unidad	Condición de prueba
V_{ILC}	Tensión de nivel bajo del reloj de entrada	−0,3		0,45	V	
V_{IHC}	Tensión de nivel alto de reloj de entrada	$V_{CC}-0,6$		$V_{CC}+0,3$	V	
V_{IL}	Tensión de nivel bajo a la entrada	−0,3		0,8	V	
V_{IH}	Tensión de nivel alto a la entrada	2,0		V_{CC}	V	
V_{OL}	Tensión de nivel bajo a la salida			0,4	V	$I_{OL} = 1,8$ mA
V_{OH}	Tensión de nivel alto a la salida	2,4			V	$I_{OH} = -250\ \mu$A
I_{CC}	Corriente de la fuente de alimentación			150	mA	
I_{LI}	Corriente de fuga a la entrada			10	μA	$V_{IN} = 0$ a V_{CC}
I_{LOH}	Corriente de fuga a la salida triestado flotante			10	μA	$V_{OUT} = 2,4$ a V_{CC}
I_{LOL}	Corriente de fuga a la salida triestado flotante			−10	μA	$V_{OUT} = 0,4$ V
I_{LD}	Corriente de fuga en el bus de datos en el modo de entrada			±10	μA	$0 \leq V_{IN} \leq V_{CC}$

CAPACITANCIA
$T_A = 25$ °C, $f = 1$ MHz,
con los terminales no medidos puestos a tierra.

Símbolo	Parámetro	Máx.	Unidad
C_Φ	Capacitancia del reloj	35	pF
C_{IN}	Capacitancia de entrada	5	pF
C_{OUT}	Capacitancia de salida	10	pF

CPU-Z80
INFORMACION PARA PEDIDOS

C - Cerámico
P - Plástico
S - Normalizado 5 V ± 5 %, de 0 °C a 70 °C
E - Ampliado 5 V ± 5 %, de −40 °C a 85 °C
M - Militar 5 V ± 10 %, de −55 °C a 125 °C

Impreso con permiso de Zilog, Inc. © 1977.

CARACTERISTICAS EN C.C. DE LA CPU-Z80A

$T_A = 0°C$ a $70°C$, $V_{CC} = 5\ V \pm 5\ \%$, salvo que se indique otra cosa.

Símbolo	Parámetro	Mín.	Típ.	Máx.	Unidad	Condición de prueba
V_{ILC}	Tensión de nivel bajo del reloj de entrada	−0,3		0,45	V	
V_{IHC}	Tensión de nivel alto de reloj de entrada	$V_{CC}-0,6$		$V_{CC}+0,3$	V	
V_{IL}	Tensión de nivel bajo a la entrada	−0,3		0,8	V	
V_{IH}	Tensión de nivel alto a la entrada	2,0		V_{CC}	V	
V_{OL}	Tensión de nivel bajo a la salida			0,4	V	$I_{OL} = 1,8$ mA
V_{OH}	Tensión de nivel alto a la salida	2,4			V	$I_{OH} = -250\ \mu A$
I_{CC}	Corriente de la fuente de alimentación		90	200	mA	
I_{LI}	Corriente de fuga a la entrada			10	μA	$V_{IN} = 0$ a V_{CC}
I_{LOH}	Corriente de fuga a la salida triestado flotante			10	μA	$V_{OUT} = 2,4$ a V_{CC}
I_{LOL}	Corriente de fuga a la salida triestado flotante			−10	μA	$V_{OUT} = 0,4$ V
I_{LD}	Corriente de fuga en el bus de datos en el modo de entrada			±10	μA	$0 \leqslant V_{IN} \leqslant V_{CC}$

CAPACITANCIA

$T_A = 25°C$, $f = 1$ MHz,
con los terminales no medios puestos a tierra.

Símbolo	Parámetro	Máx.	Unidad
C_Φ	Capacitancia del reloj	35	pF
C_{IN}	Capacitancia de entrada	5	pF
C_{OUT}	Capacitancia de salida	10	pF

CPU-Z80A
INFORMACION PARA PEDIDO

C - Cerámico
P - Plástico
S - Normalizado 5 V ± 5 %, de 0°C a 70°C

CARACTERISTICAS EN C.A. DE LA CPU-Z80

$T_A = 0°C$ a $70°C$, $V_{CC} = +5\ V \pm 5\ \%$, salvo que se indique otra cosa.

Señal	Símbolo	Parámetro	Mín.	Máx.	Unidad	Condición de prueba
Φ	t_c	Período de reloj	4	[12]	μseg	
	$t_{w(\Phi H)}$	Anchura de impulso de reloj, nivel alto	180	[E]	nseg	
	$t_{w(\Phi L)}$	Anchura de impulso de reloj, nivel bajo	180	2000	nseg	
	$t_{r,f}$	Tiempo de subida y bajada del impulso de reloj		30	nseg	
A_{0-15}	$t_{D(AD)}$	Retardo entre dirección y salida		145	nseg	$C_L = 50$ pF
	$t_{F(AD)}$	Retardo a flotación		110	nseg	
	t_{acm}	Dirección estabilizada antes de $\overline{MREQ}$ (ciclo de memoria)	[1]		nseg	[1] $t_{acm} = t_{w(\Phi H)} + t_f - 75$
	t_{aci}	Dirección estabilizada antes de $\overline{IORQ}$, $\overline{RD}$ o $\overline{WR}$ (ciclo de E/S)	[2]		nseg	[2] $t_{aci} = t_c - 80$
	t_{ca}	Dirección estabilizada desde $\overline{RD}$, $\overline{WR}$, $\overline{IORQ}$ o $\overline{MREQ}$	[3]		nseg	[3] $t_{ca} = t_{w(\Phi L)} + t_r - 40$
	t_{caf}	Dirección estabilizada desde $\overline{RD}$ o $\overline{WR}$ durante flotación	[4]		nseg	[4] $t_{caf} = t_{w(\Phi L)} + t_r - 60$
D_{0-7}	$t_{D(D)}$	Retardo de salida de datos		230	nseg	$C_L = 50$ pF
	$t_{F(D)}$	Retardo a flotación durante el ciclo de escritura		90	nseg	
	$t_{S\Phi(D)}$	Tiempo de establecimiento de los datos hasta el flanco ascendente del impulso de reloj durante el ciclo M1	50		nseg	[5] $t_{dcm} = t_c - 210$
	$t_{S\Phi(D)}$	Tiempo de establecimiento de los datos hasta el flanco descendente del impulso de reloj durante M2 hasta M5	60		nseg	[6] $t_{dci} = t_{w\Phi L)} + t_r - 210$
	t_{dcm}	Datos estabilizados antes de $\overline{WR}$ (ciclo de memoria)	[5]		nseg	[7] $t_{cdf} = t_{w(\Phi L)} + t_r - 80$
	t_{dci}	Datos estabilizados antes de $\overline{WR}$ (ciclo de E/S)	[6]		nseg	
	t_{cdf}	Datos estabilizados desde $\overline{WR}$	[7]			
	t_H	Cualquier tiempo de retención para el establecimiento	0		nseg	
$\overline{MREQ}$	$t_{DL\Phi(MR)}$	Retardo de $\overline{MREQ}$ desde el flanco descendente del impulso de reloj, para $\overline{MREQ}$ a nivel bajo		100	nseg	$C_L = 50$ pF
	$t_{DH\Phi(MR)}$	Retardo de $\overline{MREQ}$ desde el flanco ascendente del impulso de reloj, para $\overline{MREQ}$ a nivel alto		100	nseg	
	$t_{DH\Phi(MR)}$	Retardo de $\overline{MREQ}$ desde el flanco descendente del impulso de reloj, para $\overline{MREQ}$ a nivel alto		100	nseg	
	$t_{w(\overline{MRL})}$	Anchura del impulso, para $\overline{MREQ}$ a nivel bajo	[8]		nseg	[8] $t_{w(MRL)} = t_c - 40$
	$t_{w(\overline{MRH})}$	Anchura del impulso, para $\overline{MREQ}$ a nivel alto	[9]		nseg	[9] $t_{w(MRH)} = t_{w(\Phi H)} + t_f - 30$

Señal	Símbolo	Parámetro	Mín.	Máx.	Unidad	Condición de prueba
$\overline{IORQ}$	$t_{DL\Phi(IR)}$	Retardo de $\overline{IORQ}$ desde el flanco ascendente del impulso de reloj, para $\overline{IORQ}$ a nivel bajo		90	nseg	C_L=50 pF
	$t_{DL\bar{\Phi}(IR)}$	Retardo de $\overline{IORQ}$ desde el flanco descendente del impulso de reloj, para $\overline{IORQ}$ a nivel bajo		110	nseg	
	$t_{DH\Phi(IR)}$	Retardo de $\overline{IORQ}$ desde el flanco ascendente del impulso de reloj, para $\overline{IORQ}$ a nivel alto		100	nseg	
	$t_{DH\bar{\Phi}(IR)}$	Retardo de $\overline{IORQ}$ desde el flanco descendente del impulso de reloj, para $\overline{IORQ}$ a nivel alto		110	nseg	
$\overline{RD}$	$t_{DL\Phi(RD)}$	Retardo de $\overline{RD}$ desde el flanco ascendente del impulso de reloj, para $\overline{RD}$ a nivel bajo		100	nseg	C_L=50 pF
	$t_{DL\bar{\Phi}(RD)}$	Retardo de $\overline{RD}$ desde el flanco descendente del impulso de reloj, para $\overline{RD}$ a nivel bajo		130	nseg	
	$t_{DH\Phi(RD)}$	Retardo de $\overline{RD}$ desde el flanco ascendente del impulso de reloj, para $\overline{RD}$ a nivel alto		100	nseg	
	$t_{DH\bar{\Phi}(RD)}$	Retardo de $\overline{RD}$ desde el flanco descendente del impulso de reloj, para $\overline{RD}$ a nivel alto		110	nseg	
$\overline{WR}$	$t_{DL\Phi(WR)}$	Retardo de $\overline{WR}$ desde el flanco ascendente del impulso de reloj, para $\overline{WR}$ a nivel bajo		80	nseg	C_L=50 pF
	$t_{DL\bar{\Phi}(WR)}$	Retardo de $\overline{WR}$ desde el flanco descendente del impulso de reloj, para $\overline{WR}$ a nivel bajo		90	nseg	
	$t_{DH\Phi(WR)}$	Retardo de $\overline{WR}$ desde el flanco descendente del impulso de reloj, para $\overline{WR}$ a nivel alto		100	nseg	
	$t_{w(\overline{WRL})}$	Anchura del impulso, para $\overline{WR}$ a nivel bajo	[10]		nseg	
$\overline{M1}$	$t_{DL(M1)}$	Retardo de $\overline{M1}$ desde el flanco ascendente del impulso de reloj, para $\overline{M1}$ a nivel bajo		130	nseg	C_L=50 pF
	$t_{DH(M1)}$	Retardo de $\overline{M1}$ desde el flanco ascendente del impulso de reloj, para $\overline{M1}$ a nivel alto		130	nseg	
$\overline{RFSH}$	$t_{DL(RF)}$	Retardo de $\overline{RFSH}$ desde el flanco ascendente del impulso de reloj, para $\overline{RFSH}$ a nivel bajo		180	nseg	C_L=50 pF
	$t_{DH(RF)}$	Retardo de $\overline{RFSH}$ desde el flanco ascendente del impulso de reloj, para $\overline{RFSH}$ a nivel alto		150	nseg	
$\overline{WAIT}$	$t_{s(WT)}$	Tiempo de establecimiento de $\overline{WAIT}$ hasta el flanco descendente del impulso de reloj	70		nseg	
$\overline{HALT}$	$t_{D(HT)}$	Tiempo de retardo de $\overline{HALT}$ desde el flanco descendente del impulso de reloj		300	nseg	C_L=50 pF
$\overline{INT}$	$t_{s(IT)}$	Tiempo de establecimiento de $\overline{INT}$ hasta el flanco ascendente del impulso de reloj	80		nseg	
$\overline{NMI}$	$t_{w(\overline{NML})}$	Anchura del impulso, para $\overline{NMI}$ a nivel bajo	80		nseg	
$\overline{BUSRQ}$	$t_{s(BQ)}$	Tiempo de establecimiento de $\overline{BUSRQ}$ hasta el flanco ascendente del impulso de reloj	80		nseg	
$\overline{BUSAK}$	$t_{DL(BA)}$	Retardo de $\overline{BUSAK}$ desde el flanco ascendente del impulso de reloj, para $\overline{BUSAK}$ a nivel bajo		120	nseg	C_L=50 pF
	$t_{DH(BA)}$	Retardo de $\overline{BUSAK}$ desde el flanco descendente del impulso de reloj, para $\overline{BUSAK}$ a nivel alto		110	nseg	
$\overline{RESET}$	$t_{s(RS)}$	Tiempo de establecimiento de $\overline{RESET}$ hasta el flanco ascendente del impulso de reloj	90		nseg	
	$t_{F(C)}$	Retardo hasta flotación ($\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$ y $\overline{WR}$)		100	nseg	
	t_{mr}	Estabilización de $\overline{M1}$ antes de $\overline{IORQ}$ (acuse de interrupción)	[11]		nseg	

[10] $t_{w(\overline{WRL})}=t_c-40$

[11] $t_{mr}=2t_c+t_{w(\Phi H)}+t_f-80$

NOTAS:

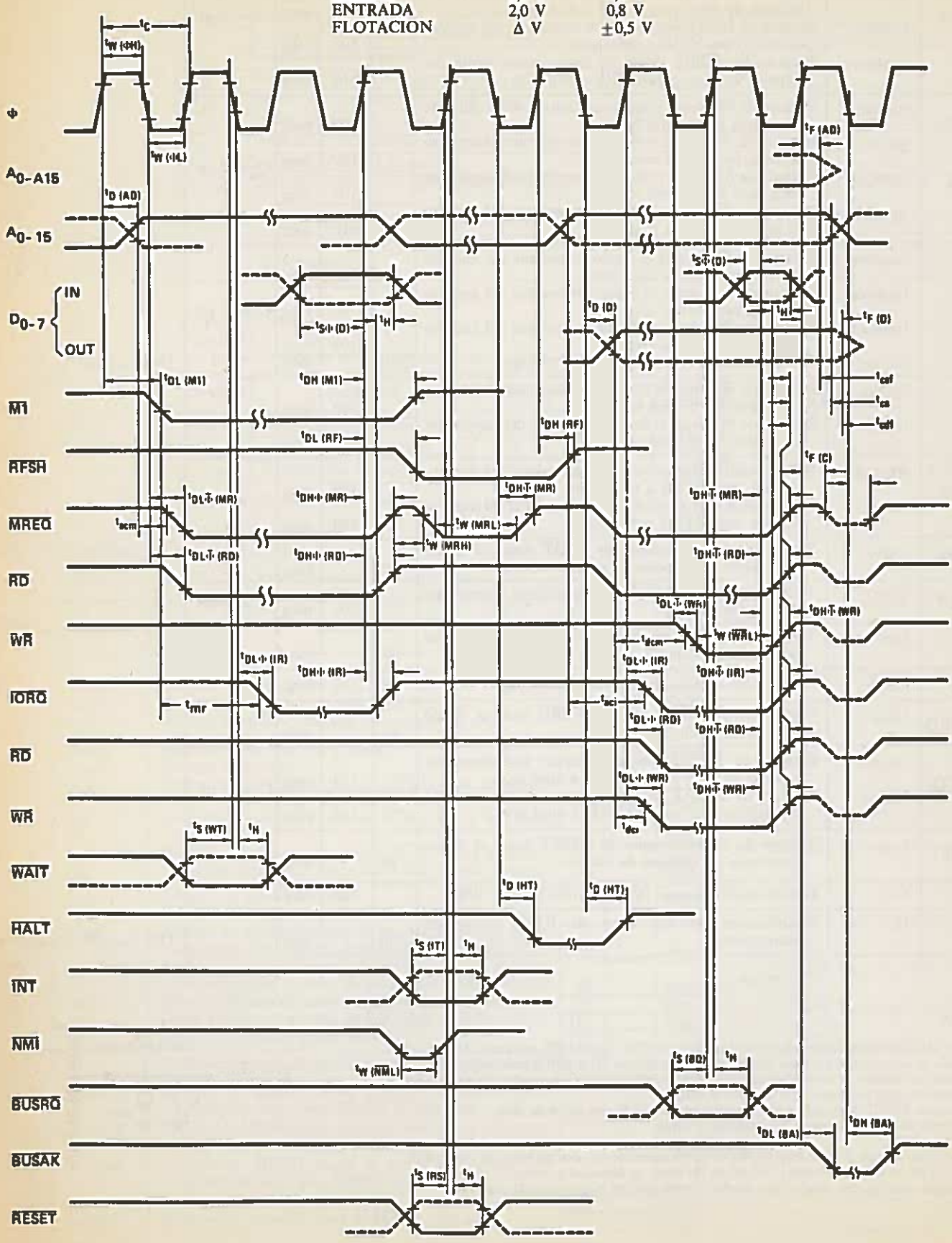
A. Los datos han de habilitarse en el bus de datos de la CPU cuando $\overline{RD}$ está activa. Durante el acuse de interrupción, los datos han de habilitarse cuando $\overline{M1}$ y $\overline{IORQ}$ están ambos activos.
B. Todas las señales de control están sincronizadas interiormente, de modo que pueden ser completamente asíncronas con respecto al reloj.
C. La señal $\overline{RESET}$ debe estar activa durante un mínimo de tres ciclos de reloj.
D. Retardo de la salida frente a capacitancia de carga.
   TA=70°C V_{CC}=+5 V±5%
   Añadir 10 nseg de retardo por cada 50 pF de aumento en la carga hasta un máximo de 200 pF en el bus de datos y 100 pF en las líneas de dirección y control.
E. Aunque estático por diseño, las pruebas garantizan un $t_{w(\Phi H)}$ de 200 µseg como máximo.

Circuito de carga para la salida

DIAGRAMA DE TIEMPOS EN C.A.

Las medidas de tiempo se han hecho con las tensiones siguientes, salvo que se indique otra cosa:

	«1»	«0»
RELOJ	$V_{CC}-0,6$ V	0,45 V
SALIDA	2,0 V	0,8 V
ENTRADA	2,0 V	0,8 V
FLOTACION	Δ V	±0,5 V



CARACTERISTICAS EN C.A. DE LA CPU-Z80

$T_A = 0\,°C$ a $70\,°C$, $V_{CC} = +5\ V \pm 5\ \%$, salvo que se indique otra cosa.

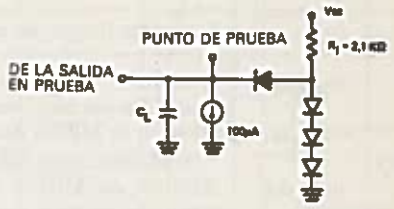
Señal	Símbolo	Parámetro	Mín.	Máx.	Unidad	Condición de prueba	
Φ	t_c	Período de reloj	25	[12]	μseg		[12] $t_c = t_{w(\Phi H)} + t_{w(\Phi L)} + t_r + t_f$
	$t_{w\,(\Phi H)}$	Anchura de impulso de reloj, nivel alto	110	[E]	nseg		
	$t_{w\,(\Phi L)}$	Anchura de impulso de reloj, nivel bajo	110	2000	nseg		
	$t_{r,\,f}$	Tiempo de subida y bajada del impulso de reloj		30	nseg		
A_{0-15}	$t_{D(AD)}$	Retardo entre dirección y salida		110	nseg	$C_L = 50$ pF	
	$t_{F(AD)}$	Retardo a flotación		90	nseg		
	t_{acm}	Dirección estabilizada antes de $\overline{MREQ}$ (ciclo de memoria)	[1]		nseg		[1] $t_{acm} = t_{w(\Phi H)} + t_f - 65$
	t_{aci}	Dirección estabilizada antes de $\overline{IORQ}$, $\overline{RD}$ o $\overline{WR}$ (ciclo de E/S)	[2]		nseg		[2] $t_{aci} = t_c - 70$
	t_{ca}	Dirección estabilizada desde $\overline{RD}$, $\overline{WR}$, $\overline{IORQ}$ o $\overline{MREQ}$	[3]		nseg		[3] $t_{ca} = t_{w(\Phi L)} + t_r - 50$
	t_{caf}	Dirección estabilizada desde $\overline{RD}$ o $\overline{WR}$ durante flotación	[4]		nseg		[4] $t_{caf} = t_{w(\Phi L)} + t_r - 45$
D_{0-7}	$t_{D(D)}$	Retardo de salida de datos		150	nseg	$C_L = 50$ pF	
	$t_{F(D)}$	Retardo a flotación durante el ciclo de escritura		90	nseg		
	$t_{S\Phi(D)}$	Tiempo de establecimiento de los datos hasta el flanco ascendente del impulso de reloj durante el ciclo M1	35		nseg		
	$t_{S\bar{\Phi}(D)}$	Tiempo de establecimiento de los datos hasta el flanco descendente del impulso de reloj durante M2 hasta M5	50		nseg		
	t_{dcm}	Datos estabilizados antes de $\overline{WR}$ (ciclo de memoria)	[5]		nseg		[5] $t_{dcm} = t_c - 170$
	t_{dci}	Datos estabilizados antes de $\overline{WR}$ (ciclo de E/S)	[6]		nseg		[6] $t_{dci} = t_{w(\Phi L)} + t_r - 170$
	t_{cdf}	Datos estabilizados desde $\overline{WR}$	[7]				[7] $t_{cdf} = t_{w(\Phi L)} + t_r - 70$
	t_H	Cualquier tiempo de retención para el establecimiento		0	nseg		
$\overline{MREQ}$	$t_{DL\bar{\Phi}(MR)}$	Retardo de $\overline{MREQ}$ desde el flanco descendente del impulso de reloj, para $\overline{MREQ}$ a nivel bajo		85	nseg	$C_L = 50$ pF	
	$t_{DH\Phi(MR)}$	Retardo de $\overline{MREQ}$ desde el flanco ascendente del impulso de reloj, para $\overline{MREQ}$ a nivel alto		85	nseg		
	$t_{DH\bar{\Phi}(MR)}$	Retardo de $\overline{MREQ}$ desde el flanco descendente del impulso de reloj, para $\overline{MREQ}$ a nivel alto		85	nseg		
	$t_{w(\overline{MRL})}$	Anchura del impulso, para $\overline{MREQ}$ a nivel bajo	[8]		nseg		[8] $t_{w(\overline{MRL})} = t_c - 30$
	$t_{w(\overline{MRH})}$	Anchura del impulso, para $\overline{MREQ}$ a nivel alto	[9]		nseg		[9] $t_{w(\overline{MRH})} = t_{w(\Phi H)} + t_f - 20$
$\overline{IORQ}$	$t_{DL\Phi(IR)}$	Retardo de $\overline{IORQ}$ desde el flanco ascendente del impulso de reloj, para $\overline{IORQ}$ a nivel bajo		75	nseg	$C_L = 50$ pF	
	$t_{DL\bar{\Phi}(IR)}$	Retardo de $\overline{IORQ}$ desde el flanco descendente del impulso de reloj, para $\overline{IORQ}$ a nivel bajo		85	nseg		
	$t_{DH\Phi(IR)}$	Retardo de $\overline{IORQ}$ desde el flanco ascendente del impulso de reloj, para $\overline{IORQ}$ a nivel alto		85	nseg		
	$t_{DH\bar{\Phi}(IR)}$	Retardo de $\overline{IORQ}$ desde el flanco descendente del impulso de reloj, para $\overline{IORQ}$ a nivel alto		85	nseg		
$\overline{RD}$	$t_{DL\Phi(RD)}$	Retardo de $\overline{RD}$ desde el flanco ascendente del impulso de reloj, para $\overline{RD}$ a nivel bajo		85	nseg	$C_L = 50$ pF	
	$t_{DL\bar{\Phi}(RD)}$	Retardo de $\overline{RD}$ desde el flanco descendente del impulso de reloj, para $\overline{RD}$ a nivel bajo		95	nseg		
	$t_{DH\Phi(RD)}$	Retardo de $\overline{RD}$ desde el flanco ascendente del impulso de reloj, para $\overline{RD}$ a nivel alto		85	nseg		
	$t_{DH\bar{\Phi}(RD)}$	Retardo de $\overline{RD}$ desde el flanco descendente del impulso de reloj, para $\overline{RD}$ a nivel alto		85	nseg		
$\overline{WR}$	$t_{DL\Phi(WR)}$	Retardo de $\overline{WR}$ desde el flanco ascendente del impulso de reloj, para $\overline{WR}$ a nivel bajo		65	nseg	$C_L = 50$ pF	
	$t_{DL\bar{\Phi}(WR)}$	Retardo de $\overline{WR}$ desde el flanco descendente del impulso de reloj, para $\overline{WR}$ a nivel bajo		80	nseg		
	$t_{DH\Phi(WR)}$	Retardo de $\overline{WR}$ desde el flanco descendente del impulso de reloj, para $\overline{WR}$ a nivel alto		80	nseg		
	$t_{w(\overline{WRL})}$	Anchura del impulso, para $\overline{WR}$ a nivel bajo	[10]		nseg		[10] $t_{w(\overline{WRL})} = t_c - 30$
$\overline{MI}$	$t_{DL(MI)}$	Retardo de $\overline{MI}$ desde el flanco ascendente del impulso de reloj, para $\overline{MI}$ a nivel bajo		100	nseg	$C_L = 50$ pF	
	$t_{DH(MI)}$	Retardo de $\overline{MI}$ desde el flanco ascendente del impulso de reloj, para $\overline{MI}$ a nivel alto		100	nseg		
$\overline{RFSH}$	$t_{DL(RF)}$	Retardo de $\overline{RFSH}$ desde el flanco ascendente del impulso de reloj, para $\overline{RFSH}$ a nivel bajo		130	nseg	$C_L = 50$ pF	
	$t_{DH(RF)}$	Retardo de $\overline{RFSH}$ desde el flanco ascendente del impulso de reloj, para $\overline{RFSH}$ a nivel alto		120	nseg		

Señal	Símbolo	Parámetro	Mín.	Máx.	Unidad	Condición de prueba
$\overline{WAIT}$	$t_{s(WT)}$	Tiempo de establecimiento de $\overline{WAIT}$ hasta el flanco descendente del impulso de reloj	70		nseg	
$\overline{HALT}$	$t_{D(HT)}$	Tiempo de retardo de $\overline{HALT}$ desde el flanco descendente del impulso de reloj		300	nseg	$C_L = 50$ pF
$\overline{INT}$	$t_{s(IT)}$	Tiempo de establecimiento de $\overline{INT}$ hasta el flanco ascendente del impulso de reloj	80		nseg	
$\overline{NMI}$	$t_{w(\overline{NML})}$	Anchura del impulso, para $\overline{NMI}$ a nivel bajo	80		nseg	
$\overline{BUSRQ}$	$t_{s(BQ)}$	Tiempo de establecimiento de $\overline{BUSRQ}$ hasta el flanco ascendente del impulso de reloj	50		nseg	
$\overline{BUSAK}$	$t_{DL(BA)}$	Retardo de $\overline{BUSAK}$ desde el flanco ascendente del impulso de reloj, para $\overline{BUSAK}$ a nivel bajo		100	nseg	$C_L = 50$pF
	$t_{DH(BA)}$	Retardo de $\overline{BUSAK}$ desde el flanco descendente del impulso de reloj, para $\overline{BUSAK}$ a nivel alto		100	nseg	
$\overline{RESET}$	$t_{s(RS)}$	Tiempo de establecimiento de $\overline{RESET}$ hasta el flanco ascendente del impulso de reloj	60		nseg	
	$t_{F(C)}$	Retardo hasta flotación ($\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$ y $\overline{WR}$)		80	nseg	
	t_{mr}	Estabilización de $\overline{M1}$ antes de $\overline{IORQ}$ (acuse de interrupción)	[11]		nseg	

[11] $t_{mr} = 2t_c + t_{w(\Phi H)} + t_f - 65$

NOTAS:

A. Los datos han de habilitarse en el bus de datos de la CPU cuando $\overline{RD}$ está activa. Durante el acuse de interrupción, los datos han de habilitarse cuando $\overline{M1}$ y $\overline{IORQ}$ están ambos activos.
B. Todas las señales de control están sincronizadas interiormente, de modo que pueden ser completamente asíncronas con respecto al reloj.
C. La señal $\overline{RESET}$ debe estar activa durante un mínimo de tres ciclos de reloj.
D. Retardo de la salida frente a capacitancia de carga.
   TA = 70 °C $V_{CC} = +5$ V ± 5 %
   Añadir 10 nseg de retardo por cada 50 pF de aumento en la carga hasta un máximo de 200 pF en el bus de datos y 100 pF en las líneas de dirección y control.
E. Aunque estático por diseño, las pruebas garantizan un $t_{w(\Phi H)}$ de 200 µseg como máximo.



Circuito de carga para la salida

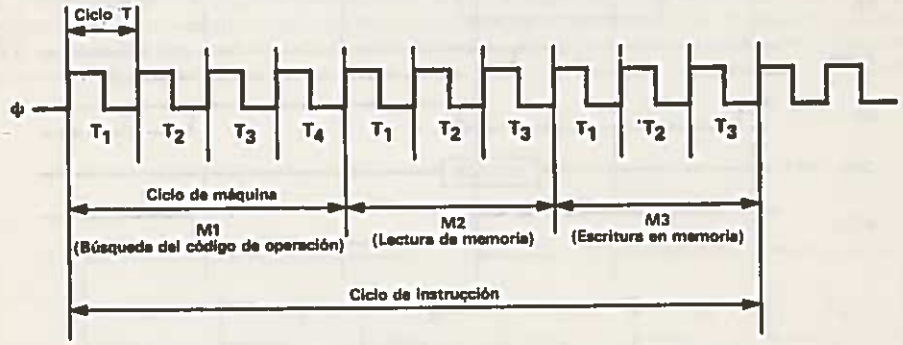
Apéndice E2

Temporización de la CPU

La CPU del Z80 ejecuta las instrucciones recorriendo un conjunto muy preciso de unas pocas operaciones básicas. Estas consisten en:

Lectura o escritura de memoria
Lectura o escritura de dispositivo de E/S
Acuse de recibo de interrupciones

Todas las instrucciones son meramente una serie de estas operaciones básicas, cada una de las cuales puede durar entre tres y seis períodos de reloj para completarse, o bien pueden alargarse con el objeto de sincronizar la CPU con la velocidad de los dispositivos exteriores. Los períodos básicos del reloj se denominan ciclos T y las operaciones básicas se denominan ciclos M (de máquina). La figura 0 muestra cómo una instrucción típica puede considerarse como una serie de ciclos específicos M y T. Adviértase que esta instrucción consta de tres ciclos de máquina (M1, M2 y M3). El primer ciclo de máquina de cualquier instrucción es un ciclo de búsqueda que dura cuatro, cinco o seis ciclos T (si no es alargado por la señal de espera que se describirá completamente en el apartado siguiente). El ciclo de búsqueda (M1) se usa para ir a buscar el código de operación (OP) de la próxima instrucción a ejecutar. Los ciclos de máquina siguientes trasladan datos entre la CPU y la memoria a los dispositivos de E/S y pueden durar cualquier número de ciclos T entre tres y cinco (de nuevo puede haber alargamiento debido a los estados de espera que sincronizan los dispositivos exteriores con la CPU). Los párrafos siguientes describen la secuencia de tiempo que ocurre dentro de cualquier ciclo básico de máquina.



EJEMPLO DE SINCRONIZACION BASICA DE LA CPU

FIGURA 0

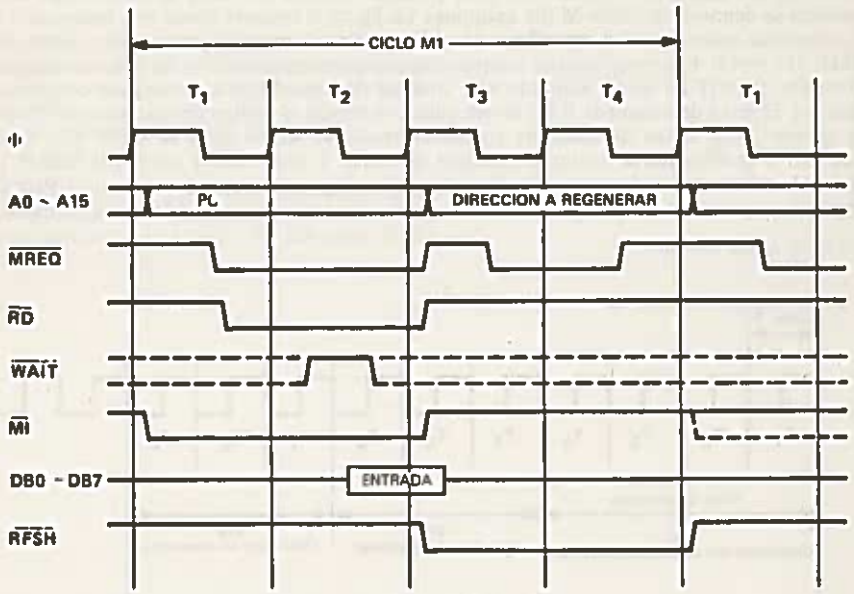
Toda la temporización de la CPU puede desglosarse en unos pocos diagramas muy simples de tiempo, tal y como se ve en las figuras 1 y 7. Estos diagramas muestran las siguientes operaciones básicas con y sin estados de espera (que se añaden para sincronizar la CPU con la memoria lenta o con los dispositivos E/S).

1. Búsqueda del código de operación de la instrucción (ciclo M1).
2. Ciclos de lectura o escritura de datos en la memoria.
3. Ciclos de lectura o escritura de E/S.
4. Ciclo de Solicitud/Acuse del bus.
5. Ciclo de Solicitud/Acuse de interrupción.
6. Ciclo de Solicitud/Acuse de interrupción no enmascarable.
7. Salida de la instrucción HALT.

Impreso con permiso de Zilog, Inc. © 1977.

BUSQUEDA DE INSTRUCCION

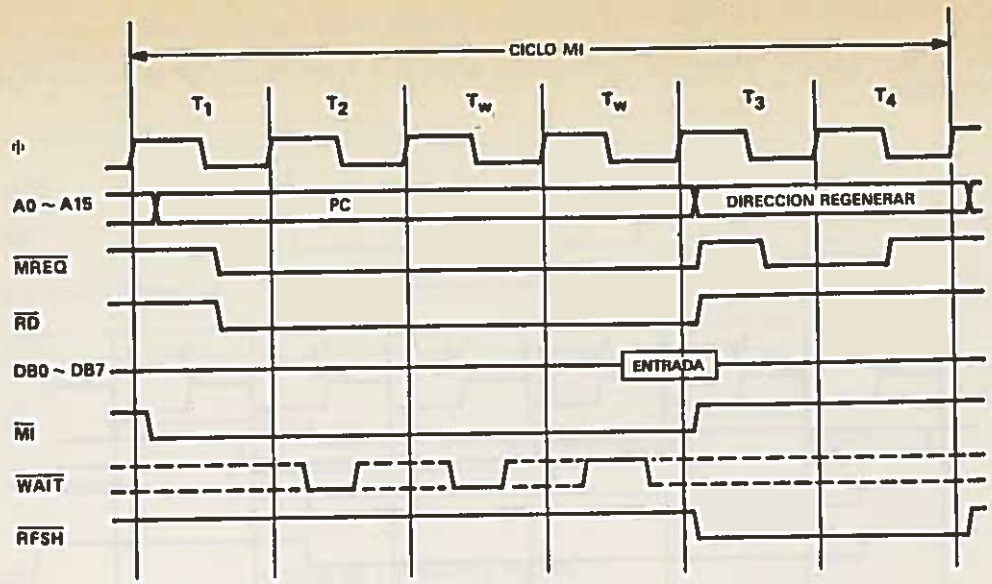
La figura 1 muestra los cronogramas durante un ciclo M1 (búsqueda del código de operación). Adviértase que el PC se coloca en el bus de direcciones al principio del ciclo M1. Medio tiempo de reloj más tarde, la señal $\overline{MREQ}$ se hace activa. En este momento, la dirección ha tenido tiempo de estabilizarse en la memoria de modo que el flanco descendente de $\overline{MREQ}$ se puede usar directamente como señal de reloj de habilitación de chip en las memorias dinámicas. La línea $\overline{RD}$ se hace también activa para indicar que hay que habilitar el paso de los datos leídos en la memoria al bus de datos de la CPU. La CPU muestra los datos de la memoria que están en el bus en el momento en que crece el flanco del impulso de reloj correspondiente al estado T3 y este mismo flanco lo usa la CPU para desactivar las señales $\overline{RD}$ y $\overline{MREQ}$. De este modo, los datos han sido muestreados por la CPU antes de que la señal $\overline{RD}$ se inactive. Los estados T3 y T4 del reloj en un ciclo de búsqueda se usan para regenerar las memorias dinámicas. (La CPU emplea este tiempo para decodificar y ejecutar la instrucción buscada, de modo que no es posible ejecutar ninguna otra operación durante este tiempo.) Durante T3 y T4 los 7 bits menos significativos del bus de direcciones contienen una dirección de regeneración de memoria y la señal $\overline{RFSH}$ se hace activa indicando que debe realizarse una lectura de regeneración de todas las memorias dinámicas. Nótese que durante el lapso de regeneración no se produce la señal $\overline{RD}$ para impedir que lleguen al bus datos de segmentos diferentes de la memoria. La señal $\overline{MREQ}$ debe usarse durante el lapso de regeneración para llevar a cabo la lectura de regeneración en todos los elementos de memoria. La señal de regeneración misma no se puede usar para ello, ya que la estabilidad de la dirección a regenerar sólo se garantiza durante el tiempo de $\overline{MREQ}$.



BUSQUEDA DEL CODIGO DE OPERACION DE LA INSTRUCCION

FIGURA 1

La figura 1A muestra cómo se retarda el ciclo de búsqueda si la memoria activa la línea $\overline{WAIT}$. Durante T2 y los siguientes Tw, la CPU muestrea la línea $\overline{WAIT}$ durante el flanco descendente de Φ. Si la línea está activa en este momento, se introduce otro estado de espera durante el ciclo siguiente. Con este procedimiento se puede alargar el ciclo de lectura para acoplarlo al tiempo de acceso de cualquier tipo de dispositivo de memoria.

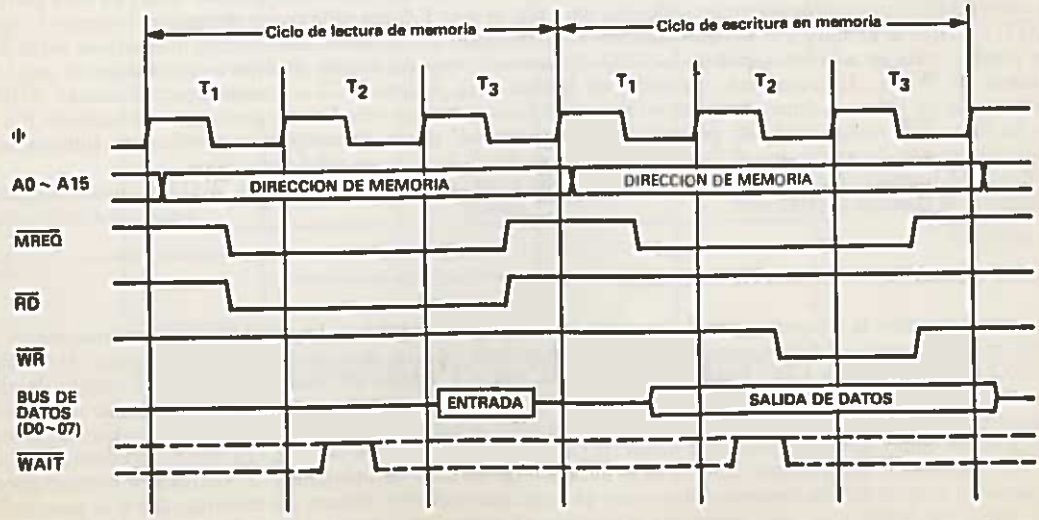


BUSQUEDA DEL CODIGO DE OPERACION DE LA INSTRUCCION CON ESTADOS DE ESPERA

FIGURA 1A

LECTURA O ESCRITURA DE MEMORIA

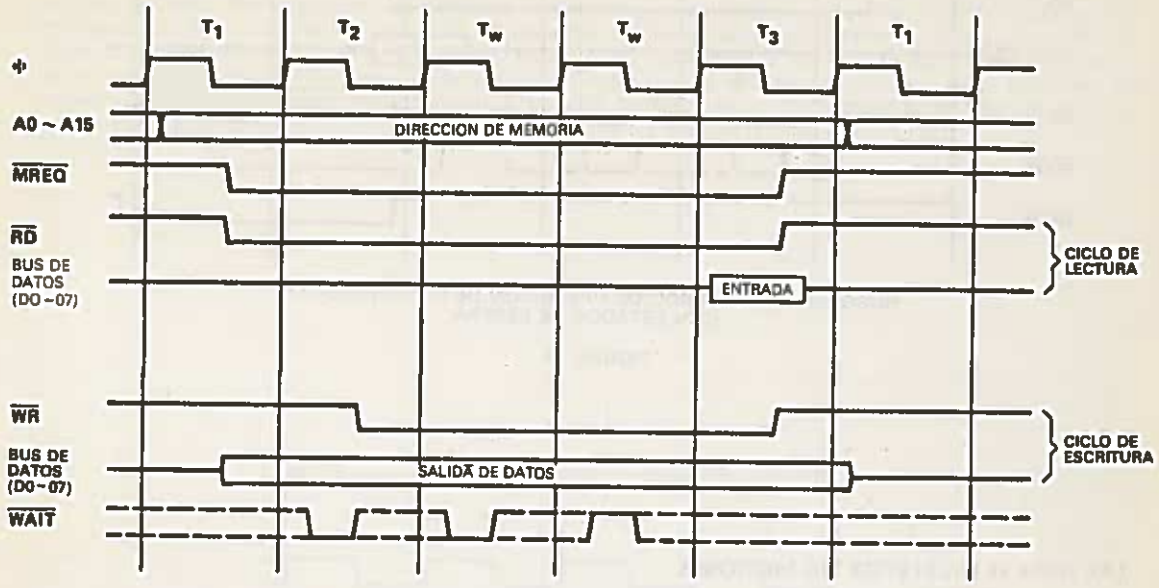
La figura 2 muestra los cronogramas de los ciclos de lectura o escritura de memoria, diferentes del de búsqueda del código de operación (ciclo M1). Estos ciclos suelen durar tres periodos de reloj, salvo que la memoria requiera estados de espera mediante la señal WAIT. Las señales MREQ y RD se usan como en el ciclo de búsqueda. En el caso de un ciclo de escritura en memoria, la MREQ se vuelve activa también cuando el bus de direcciones se estabiliza y así se la puede usar directamente como señal de habilitación de chip en las memorias dinámicas. La línea WR se activa cuando los datos se estabilizan en el bus de datos, de modo que se la puede usar directamente como impulso R/W (lectura/escritura) prácticamente con cualquier tipo de memoria de semiconductores. Más aún, la señal WR se torna inactiva medio estado T antes de que el contenido del bus de direcciones y datos se cambie y así se puedan cumplir los requisitos de solapamiento de casi todos los tipos de memorias de semiconductores.



CICLOS DE LECTURA O ESCRITURA EN MEMORIA

FIGURA 2

La figura 2A muestra cómo una señal $\overline{WAIT}$ de solicitud de espera alarga una operación de lectura o escritura. Este funcionamiento es idéntico al descrito previamente para el ciclo de búsqueda. Nótese que en esta figura se muestra un ciclo separado de lectura y otro separado de escritura, si bien ambos no pueden ocurrir nunca simultáneamente.



CICLOS DE LECTURA O ESCRITURA EN MEMORIA CON ESTADOS DE ESPERA

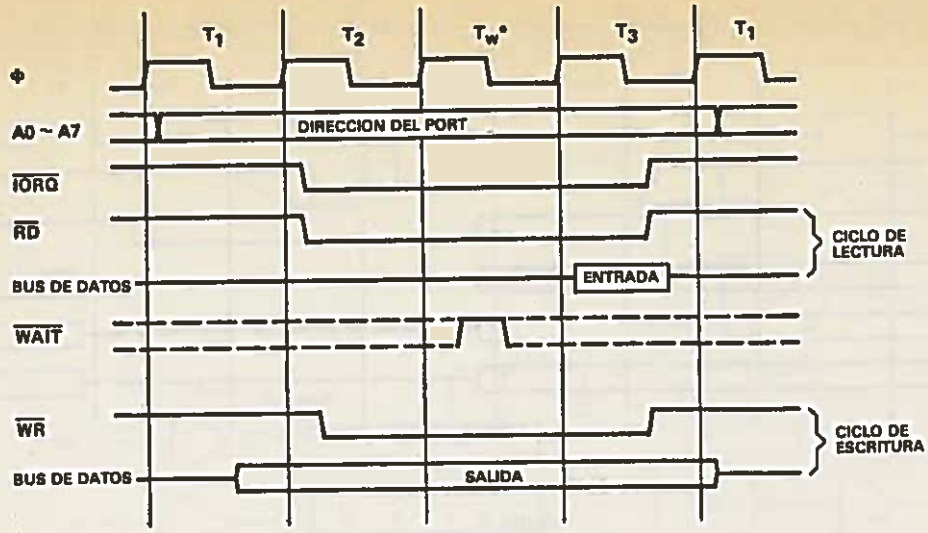
FIGURA 2A

CICLOS DE ENTRADA O SALIDA

La figura 3 muestra una operación de lectura de E/S o escritura en E/S. Nótese que durante las operaciones E/S se inserta automáticamente un estado de espera sencillo. La razón consiste en que durante estas operaciones E/S, el tiempo desde que la señal $\overline{IORQ}$ se hace activa hasta que la CPU debe muestrear la línea $\overline{WAIT}$ es muy corto y sin este estado extra no habría tiempo suficiente para que el port E/S decodificara su dirección y activara la línea $\overline{WAIT}$, si es que se necesita una espera. También sin este estado de espera es difícil diseñar dispositivos MOS E/S que puedan trabajar a la velocidad de la CPU. Durante el lapso del estado de espera se muestrea la señal de solicitud de $\overline{WAIT}$. Durante una operación de lectura E/S, la línea $\overline{RD}$ se emplea para habilitar el port direccionado en el bus de datos igual que en el caso de lectura de memoria. En las operaciones de escritura E/S se usa la línea $\overline{WR}$ como reloj del port de E/S, otra vez con plazo de solapamiento suficiente suministrado automáticamente, de modo que el flanco ascendente se pueda usar como reloj para datos.
La figura 3A muestra cómo se pueden añadir estados de espera adicionales con la línea $\overline{WAIT}$. El funcionamiento es idéntico al descrito previamente.

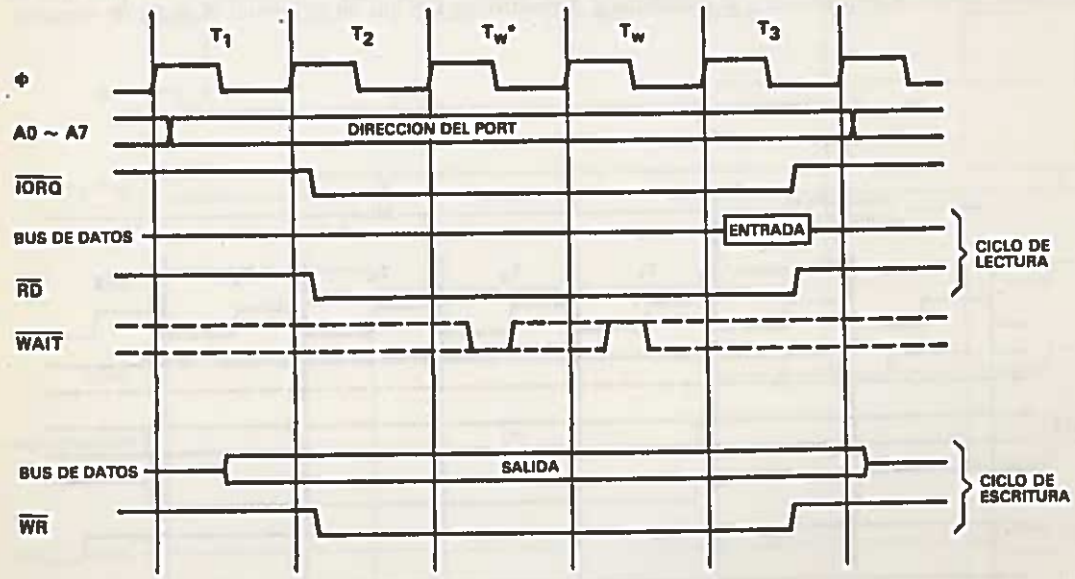
CICLO DE SOLICITUD/ACUSE DEL BUS

La figura 4 muestra la temporización de un ciclo de solicitud/acuse del bus. La señal $\overline{BUSRQ}$ se muestrea en la CPU, en el momento del flanco ascendente del último período de reloj de cualquier ciclo de máquina. Si la señal $\overline{BUSRQ}$ está activada, la CPU establece su dirección, datos y señales de control triestados al estado de alta impedancia, al tiempo del flanco ascendente del siguiente impulso de reloj. En este momento, cualquier dispositivo exterior puede controlar los buses para transferir datos entre la memoria y los dispositivos de E/S. (Esto se conoce generalmente como Acceso Directo a la Memoria [ADM] con eliminación de ciclo.) El tiempo máximo para que la CPU responda a una solicitud del bus es la duración de un ciclo de máquina y el controlador externo puede mantener su control del bus durante tantos ciclos de reloj como necesite. Nótese, sin embargo, que si se usan ciclos muy largos de ADM y se usan memorias dinámicas, el controlador externo tiene que realizar la función de regeneración. Esta situación se da únicamente cuando se transfieren bloques de datos muy grandes bajo control de ADM. Nótese también que durante el ciclo de solicitud del bus no se puede interrumpir a la CPU con ninguna señal $\overline{NMI}$ ni $\overline{INT}$.



CICLOS DE ENTRADA O SALIDA

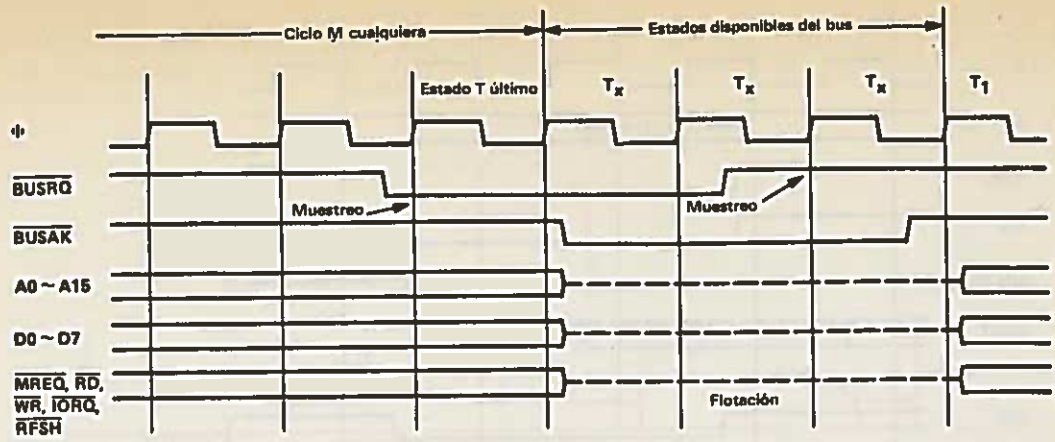
FIGURA 3



CICLOS DE ENTRADA O SALIDA CON ESTADOS DE ESPERA

FIGURA 3A

* Estado WAIT insertado automáticamente.

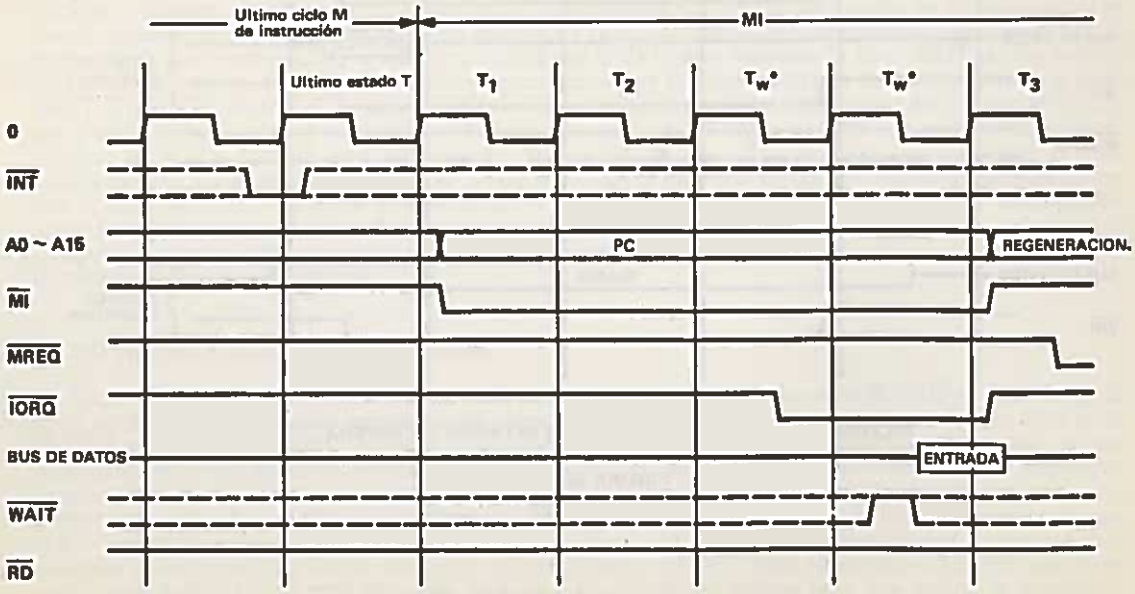


CICLO DE SOLICITUD/ACUSE DEL BUS

FIGURA 4

CICLO DE SOLICITUD/ACUSE DE INTERRUPCION

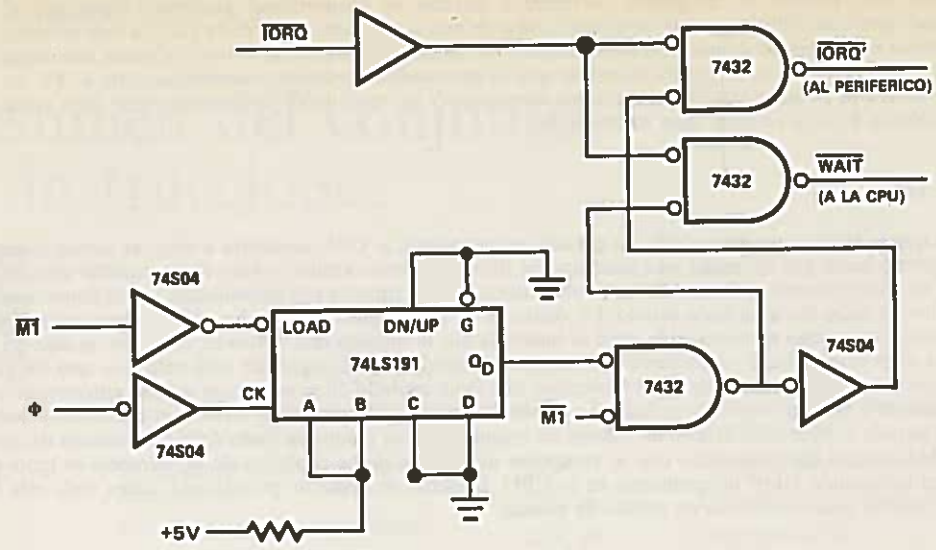
La figura 5 muestra los cronogramas asociados con un ciclo de interrupción. La CPU muestrea la señal de interrupción (INT) con el flanco ascendente del último impulso de reloj al final de cada instrucción. La señal no es aceptada si el biestable que habilita las interrupciones y que está controlado por la programación interna de la CPU no está puesto a «uno», o si la señal BUSRQ está activada. Cuando la señal resulta aceptada, se genera un ciclo M1 especial durante el cual la señal IORQ se vuelve activa (en lugar de la MREQ normal) indicando que el dispositivo interruptor puede colocar un vector de 8 bits en el bus de datos. Nótese que se añaden automáticamente dos estados de espera a este ciclo para que se pueda llevar a cabo fácilmente el esquema de interrupciones de prioridad en serie. Ambos estados de espera introducen el tiempo suficiente para que se estabilicen las señales de prioridad y se identifique el dispositivo de E/S que ha de insertar el vector de respuesta.



CICLO DE SOLICITUD/ACUSE DE INTERRUPCION

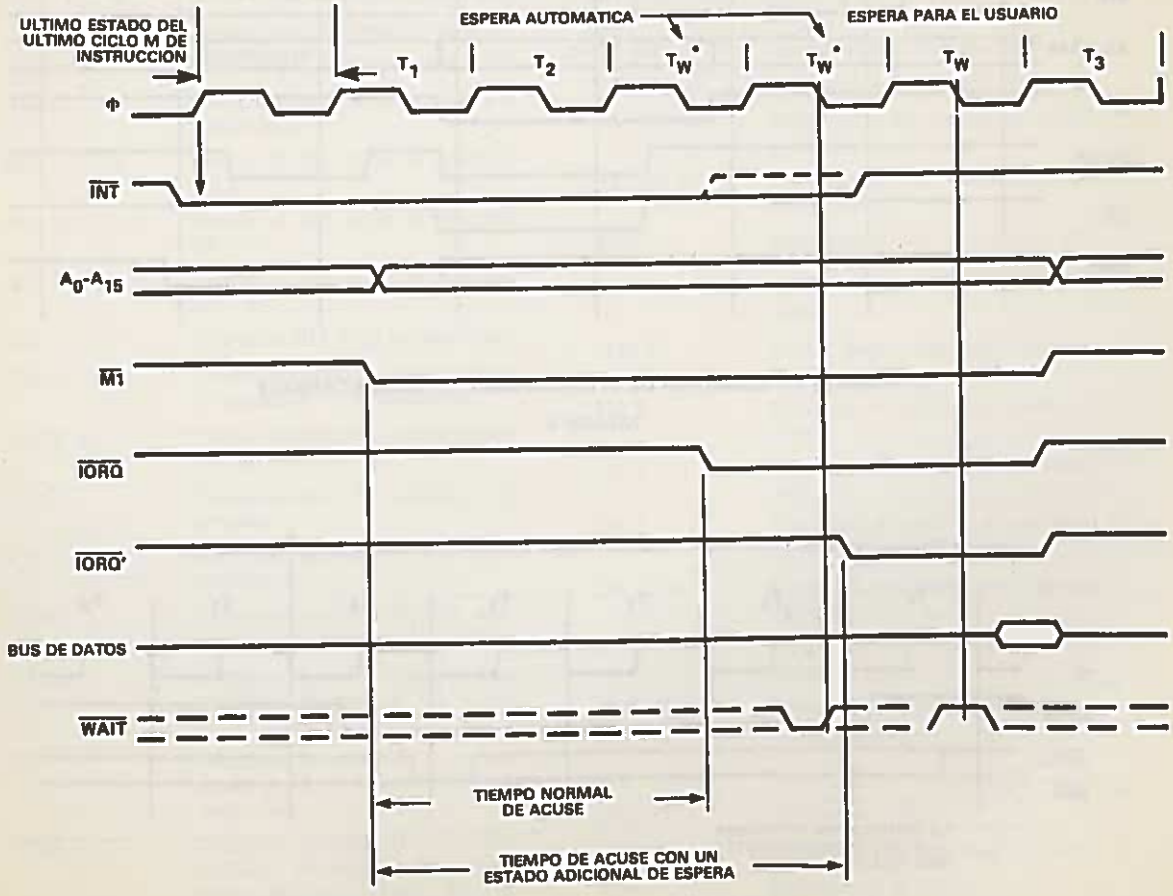
FIGURA 5

Las figuras 5A y 5B muestran cómo hay que usar un contador programable para ampliar el tiempo de acuse de interrupción. (Configurado como se muestra para añadir un estado de espera.)



AMPLIACION DEL TIEMPO DE ACUSE DE INTERRUPCIONES MEDIANTE ESTADO DE ESPERA

FIGURA 5A



CICLO DE SOLICITUD/ACUSE CON ESTADO ADICIONAL DE ESPERA

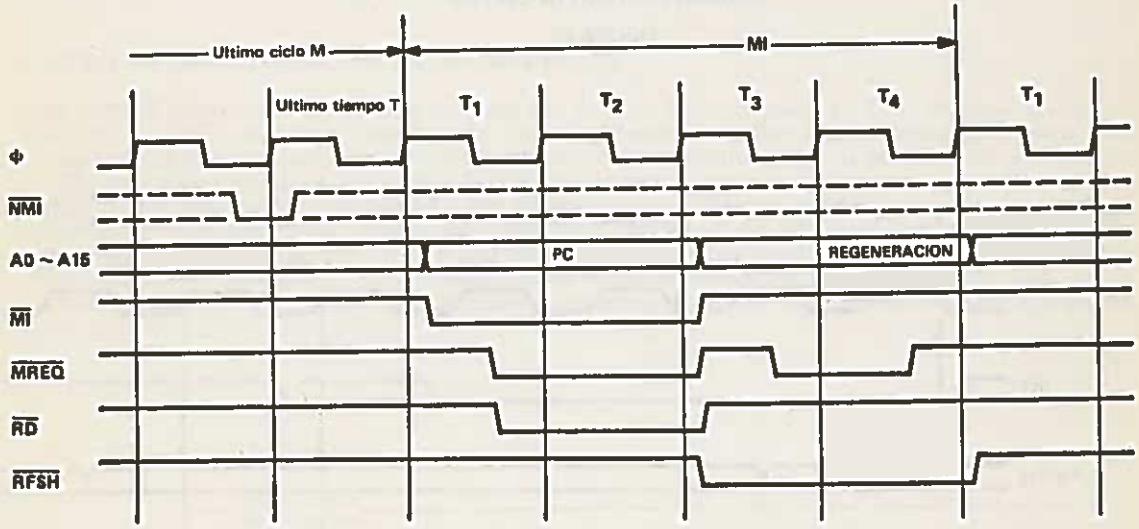
FIGURA 5B

RESPUESTA A INTERRUPCION NO ENMASCARABLE

La figura 6 muestra el ciclo de solicitud/acuse para una interrupción no enmascarable. Esta señal se muestrea al mismo tiempo que la línea de interrupciones, pero tiene prioridad sobre la interrupción normal y no puede ser inhabilitada bajo control de programa. Su función normal es proporcionar respuesta inmediata a señales importantes como un fallo inminente en el suministro de energía. La respuesta de la CPU a una interrupción no enmascarable es semejante a una operación normal de lectura de memoria; siendo la única diferencia que el contenido del bus de datos se ignora mientras que el procesador almacena automáticamente el PC en la pila externa y salta a la posición 0066_H. La rutina de servicio para la interrupción no enmascarable debe comenzar en esta posición, si se va a emplear esta interrupción.

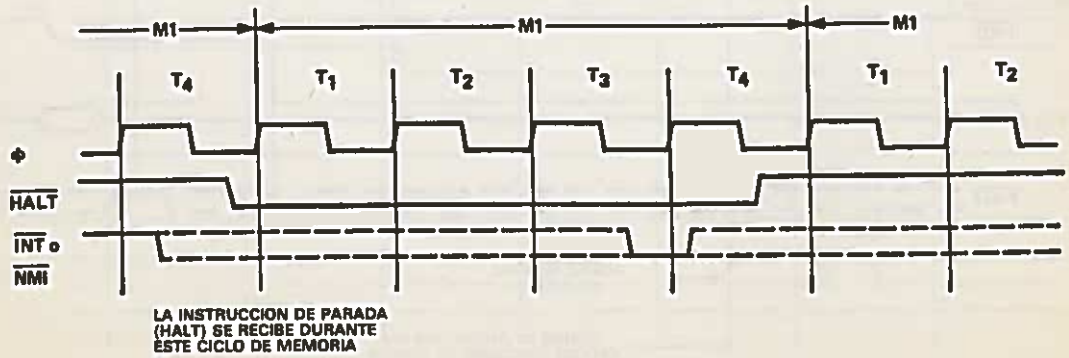
SALIDA DE HALT

Cada vez que se ejecuta una instrucción de parada del programa, la CPU comienza a ejecutar instrucciones NOP (sin operación) hasta que se recibe una interrupción (bien no enmascarable, o bien enmascarable siempre que el biestable de interrupciones esté habilitado). Ambas líneas de interrupción son examinadas con el flanco ascendente del impulso de reloj, durante cada estado T4, como se ve en la figura 7. Si se ha recibido una interrupción no enmascarable o si es una enmascarable, pero el biestable que la habilita está activado, entonces se sale del estado de parada al próximo flanco ascendente del impulso de reloj. El ciclo siguiente será entonces uno de acuse de interrupción correspondiente al tipo de interrupción que se ha recibido. Si se recibieron ambas, entonces se acusa la no enmascarable por su mayor prioridad. La razón de ejecutar instrucciones NOP (sin operación) durante el estado de parada es mantener activas las señales de regeneración de memoria. Cada ciclo en el estado de parada es un ciclo M1 normal (de búsqueda), con la excepción de que los datos recibidos de la memoria se ignoran y se fuerza otra instrucción NOP internamente en la CPU. La señal de acuse de parada está activa todo este tiempo indicando que el procesador está en estado de parada.



OPERACION DE SOLICITUD DE INTERRUPCION NO ENMASCARABLE

FIGURA 6



SALIDA DE HALT

FIGURA 7

Apéndice E3

Resumen del conjunto de instrucciones

ADC HL, ss	Suma con arrastre del par de Registros ss a HL.
ADC A, s	Suma con arrastre operando s al Acumulador.
ADD A, n	Suma el valor n al Acumulador.
ADD A, r	Suma el Registro r al Acumulador.
ADD A, (HL)	Suma la posición (HL al Acumulador.
ADD A, (IX+d)	Suma la posición (IX+d) al Acumulador.
ADD A, (IY+d)	Suma la posición (IY+d) al Acumulador.
ADD HL, ss	Suma el par de Registros ss a HL.
ADD IX, pp	Suma el par de Registros pp a IX.
ADD IY, rr	Suma el par de Registros rr a IY.
AND s	«AND» lógica de operandos s y Acumulador.
BIT b, (HL)	Prueba el BIT b de la posición (HL).
BIT b, (IX+d)	Prueba el BIT b de la posición (IX+d).
BIT b, (IY+d)	Prueba el BIT b de la posición (IY+d).
BIT b, r	Prueba el BIT b de la posición r.
CALL cc, nn	Llama subrutina en posición nn si la condición cc es verdadera.
CALL nn	Llama incondicionalmente subrutina en posición nn.
CCF	Complementa el indicador de arrastre.
CP s	Compara operando s con Acumulador.
CPD	Compara posición (HL) y Acumulador, decrementa HL y BC.
CPDR	Compara posición (HL) y Acumulador, decrementa HL y BC, repetir hasta BC=0.
CPI	Compara posición (HL) y Acumulador, incrementa HL y decrementa BC.
CPIR	Compara posición (HL) y Acumulador, incrementa HL, decrementa BC, repetir hasta BC=0.
CPL	Complementa Acumulador (complemento a 1).
DAA	Ajuste decimal del Acumulador.
DEC m	Decrementa operando m.
DEC IX	Decrementa IX.
DEC IY	Decrementa IY.
DEC ss	Decrementa el par de Registros ss.
DI	Inhabilita interrupciones.
DJNZ e	Decrementa B y salto relativo si B≠0.
EI	Habilita interrupciones.
EX (SP), HL	Intercambia la posición (SP) y HL.
EX (SP), IX	Intercambia la posición (SP) e IX.
EX (SP), IY	Intercambia la posición (SP) e IY.
EX AF, AF'	Intercambia los contenidos de AF y AF'.
EX DE, HL	Intercambia los contenidos de DE y HL.
EXX	Intercambia los contenidos de BC, DE, HL con los contenidos de BC', DE', HL', respectivamente.
HALT	HALT (espera una interrupción o puesta a cero).
IM 0	Establece el modo 0 de interrupción.
IM 1	Establece el modo 1 de interrupción.
IM 2	Establece el modo 2 de interrupción.
IN A, (n)	Carga Acumulador con entrada del dispositivo n.
IN r, (C)	Carga Registro r con entrada del dispositivo (C).
INC (HL)	Incrementa la posición (HL).
INC IX	Incrementa IX.
INC (IX+d)	Incrementa la posición (IX+d).
INC IY	Incrementa IY.
INC (IY+d)	Incrementa la localidad (IY+d).
INC r	Incrementa el Registro r.

Impreso con permiso de Zilog, Inc. © 1977.

INC ss	Incrementa el par de Registros ss.
IND	Carga la posición (HL) con la entrada del port (C), decrementa HL y B.
INDR	Carga la posición (HL) con la entrada del port (C), decrementa HL y decrementa B, repetir hasta que B=0.
INI	Carga la posición (HL) con la entrada del port (C), incrementa HL y decrementa B.
INIR	Carga la posición (HL) con la entrada del port (C), incrementa HL y decrementa B, repetir hasta que B=0.
JP (HL)	Salto incondicional a (HL).
JP (IX)	Salto incondicional a (IX).
JP (IY)	Salto incondicional a (IY).
JP cc, nn	Salto a la posición nn si la condición cc es verdadera.
JP nn	Salto incondicional a la posición nn.
JP C, e	Salto relativo a PC+e si el arrastre=1.
JR e	Salto incondicional relativo a PC+e.
JP NC, e	Salto relativo a PC+e si el acarreo=0.
JR NZ, e	Salto relativo a PC+e si no es cero (Z=0).
JR Z, e	Salto relativo a PC+e si es cero (Z=1).
LD A, (BC)	Carga Acumulador con posición (BC).
LD A, (DE)	Carga Acumulador con posición (DE).
LD A, I	Carga Acumulador con I.
LD A, (nn)	Carga Acumulador con posición nn.
LD A, R	Carga Acumulador con Registro R.
LD (BC), A	Carga posición (BC) con Acumulador.
LD (DE), A	Carga posición (DE) con Acumulador.
LD (HL), n	Carga posición (HL) con el valor n.
LD dd, nn	Carga el par de Registros dd con el valor n.
LD HL, (nn)	Carga HL con posición (nn).
LD (HL), r	Carga posición (HL) con Registro r.
LD I, A	Carga I con Acumulador.
LF IX, nn	Carga IX con el valor nn.
LD IX, (nn)	Carga IX con la posición (nn).
LD (IX+d), n	Carga posición (IX+d) con el valor n.
LD (IX+d), r	Carga posición (IX+d) con el Registro r.
LD IY, nn	Carga IY con el valor n.
LD IY, (nn)	Carga IY con la posición (nn).
LD (IY+d), n	Carga posición (IY+d) con el valor n.
LD (IY+d), r	Carga posición (IY+d) con el Registro r.
LD (nn), A	Carga posición (nn) con Acumulador.
LD (nn), dd	Carga posición (nn) con el par de Registros dd.
LD (nn), HL	Carga posición (nn) con HL.
LD (nn), IX	Carga posición (nn) con IX.
LD (nn), IY	Carga posición (nn) con IY.
LD R, A	Carga R con Acumulador.
LD r, (HL)	Carga Registro r con posición (HL).
LD r, (IX+d)	Carga Registro r con posición (IX+d).
LD r, (IY+d)	Carga Registro r con posición (IY+d).
LD r, n	Carga Registro r con el valor n.
LD r, r	Carga Registro r con Registro r'.
LD SP, HL	Carga SP con HL.
LD SP, IX	Carga SP con IX.
LD SP, IY	Carga SP con IY.
LDD	Carga posición (DE) con posición (HL), decrementa DE, HL y BC.
LDDR	Carga posición (DE) con posición (HL), decrementa DE, HL y BC, repetir hasta que BC=0.
LDI	Carga posición (DE) con posición (HL), incrementa DE y HL, decrementa BC.
LDIR	Carga posición (DE) con posición (HL), incrementa DE y HL, decrementa BC y repite hasta que BC=0.
NEG	Invierte el Acumulador (complemento a 2).
NOP	No operación.
OR s	«OR» lógica del operando s y el Acumulador.
OTDR	Carga el port (C) de salida con posición (HL), decrementa HL y B, repite hasta que B=0.
OTIR	Carga port (C) de salida con posición (HL), incrementa HL, decrementa B, repite hasta que B=0.
OUT (C), r	Carga port (C) de salida con Registro r.
OUT (n), A	Carga port (n) de salida con Acumulador.
OUTD	Carga port (C) de salida con posición (HL), decrementa HL y B.
OUTI	Carga port (C) de salida con posición (HL), incrementa HL y decrementa B.
POP IX	Carga IX con la primera fila de pila.
POP IY	Carga IY con la primera fila de pila.
POP qq	Carga el par de Registros qq en la pila.
PUSH IX	Carga IX en la pila.

PUSH IY	Carga IY en la pila.
PUSH qq	Carga el par de Registros qq en la pila.
RES b, m	Pon a cero el BIT b del operando m.
RET	Retorno de subrutina.
RET cc	Retorno de subrutina si la condición cc es verdadera.
RETI	Retorno de interrupción.
RETN	Retorno de interrupción no enmascarable.
RL m	Rotación hacia la izquierda del operando m considerando el arrastre.
RLA	Rotación hacia la izquierda del Acumulador considerando el arrastre.
RLC (HL)	Rotación circular de posición (HL) hacia la izquierda.
RLC (IX+d)	Rotación circular de posición (IX+d) hacia la izquierda.
RLC (IY+d)	Rotación circular de posición (IY+d) hacia la izquierda.
RLC r	Rotación circular del Registro r hacia la izquierda.
RLCA	Rotación circular hacia la izquierda del Acumulador.
RLD	Rotación de un dígito hacia la izquierda y hacia la derecha entre Acumulador y posición (HL).
RR m	Rotación hacia la derecha del operando m considerando el arrastre.
RRA	Rotación hacia la derecha del Acumulador considerando el arrastre.
RRC m	Rotación circular del operando m hacia la derecha.
RRCA	Rotación circular hacia la derecha del Acumulador.
RRD	Rotación de un dígito hacia la derecha y hacia la izquierda entre Acumulador y posición (HL).
RST p	Recomenzar en posición p.
SBC A, s	Restar operando s del Acumulador con arrastre.
SBC HL, ss	Restar con arrastre el par de Registros ss de HL.
SCF	Poner a «uno» el indicador de arrastre (C=1).
SET b, (HL)	Poner a «uno» el Bit b de la posición (HL).
SET b, (IX+d)	Poner a «uno» el Bit b de la posición (IX+d).
SET b, (IY+d)	Poner a «uno» el Bit b de la posición (IY+d).
SET b, r	Poner a «uno» el Bit b del Registro r.
SLA m	Desplazamiento aritmético hacia la izquierda del operando m.
SRA m	Desplazamiento aritmético hacia la derecha del operando m.
SRL m	Desplazamiento lógico hacia la derecha del operando m.
SUB s	Restar el operando s del Acumulador.
XOR s	«OR» exclusiva entre el operando s y el Acumulador.

Glosario

Acumulador. Un registro temporal en el que se pueden almacenar los resultados de los cálculos mediante la unidad central de proceso. Una unidad aritmético-lógica puede tener uno o más acumuladores.

Acoplador acústico. Un dispositivo que permite a un terminal conectarse a la computadora por medio de una línea telefónica. Se conecta al microauricular del teléfono.

Dirección. Un número de identificación o rótulo para posiciones en la memoria.

Algoritmo. Una solución paso a paso a un problema en un número finito de pasos. Un procedimiento específico para la obtención de un resultado deseado.

ASCII. Código americano estándar para intercambio de información (siglas de American Standard Code for Information Interchange). Utilizado ampliamente como código estándar de 7 bits. Se conoce también como USASCII; IBM utiliza EBCDIC que tiene 8 bits.

Ensamblador. Un programa que convierte instrucciones simbólicas en instrucciones de máquina.

Backplane (Tarjeta base). Una tarjeta provista de enchufes interconectados mediante buses en los que se pueden insertar los módulos que componen una computadora. Se conoce también como placa matriz o tarjeta base o matriz («motherboard»).

BASIC. Código simbólico de instrucciones de uso general para principiantes (siglas de Beginner's All-purpose Symbolic Instruction Code). Lenguaje algebraico desarrollado en Darmouth College. El lenguaje es fácil de aprender y usar.

Binario. Un sistema de numeración basado en múltiples de dos utilizando los dígitos 0 y 1.

Bit. Abreviatura de «binary digit» (dígito binario). Cada elemento de un número binario, bien un 0 o un 1. Los bits se representan en una microcomputadora por el estado de conmutadores electrónicos que pueden estar cerrados o abiertos. Cuatro bits son un nibble; ocho bits son un octeto (byte).

Octeto (byte). Un grupo de bits adyacentes, usualmente ocho bits, que se tratan como una unidad por el procesador central.

CMOS. Semiconductor complementario de óxido de metal (siglas de Complementary Metal-Oxide Semiconductor). Tecnología que combina la densidad de componentes de MOS canal-P (PMOS) y la velocidad de MOS canal-N (NMOS). La disipación de potencia es muy baja.

Reloj. Un dispositivo que genera impulsos regulares que sicroniza eventos a través de un microcomputador.

Unidad central de proceso. La unidad central de proceso controla la operación de una microcomputadora. La unidad central de proceso puede extraer y almacenar datos e instrucciones desde la memoria.

CRT. Tubo de rayos catódicos (siglas de Cathode-Ray Tube). Un tubo de vacío electrónico que se puede utilizar para presentaciones gráficas. Se denomina también a un terminal que incorpora un CRT. Las siglas de este dispositivo en español son TRC.

Compilador. Un programa que traduce (convierte) el lenguaje de programación de alto nivel en lenguaje máquina. Puede producir numerosas macroinstrucciones para cada instrucción de alto nivel, a diferencia con un ensamblador que traduce ítem a ítem (elemento de información). Cuando se utiliza un compilador no se puede cambiar un programa sin recompilación.

Sistema de desarrollo. Un sistema microcomputadora que tiene equipos afines necesarios para el desarrollo de hardware y software.

Digital. Relativo a magnitudes discretas en una base dada que puede expresar las variables que se presentan en un problema. Representadas electrónicamente por 2 estados (binarios) a 16 estados (hexadecimal) en el momento presente. Se distingue del analógico en que este último se refiere a una gama continua de magnitudes de tensión o corrientes.

Doble densidad. Método de duplicación de la densidad de bits en medios de almacenamiento magnético.

Memoria dinámica. Almacenamiento de datos en chips dinámicos en los que el almacenamiento de una pequeña carga indica un bit. Debido a que la carga se pierde con el tiempo, la memoria dinámica se debe regenerar («refrescar») periódicamente.

EBCDIC. Código de 8 bits de IBM, similar al ASCII.

Editor. Un programa que efectúa una nueva disposición del texto. Permite la adición o supresión de símbolos y cambios de formatos.

EIA-RS-232C. Interfaz estándar para datos transmitidos secuencialmente que no son síncronos con la unidad central de proceso.

EPROM. Memoria de sólo lectura borrable y programable (siglas de Erasable-Programmable Read-Only Memory). Una PROM que puede ser borrada y reprogramada. Algunas EPROM tienen una ventana de cuarzo sobre el chip; los datos se pueden borrar por exposición a luz intensiva ultravioleta; otras EPROM se pueden borrar electrónicamente.

Fichero. Un conjunto de registros afines tratados como una unidad.

Indicador. Un bit unido a una palabra para la identificación o con la finalidad de destacar alguna condición o estado. Los microprocesadores típicos incluyen indicadores de estado, acarreo, cero, signo, desbordamiento y semiacarreo.

Paquete de coma flotante. Un conjunto de rutinas software que permite a algunas microcomputadoras ejecutar aritmética de coma flotante sin la adición de hardware extra.

FSK. Modulación por desplazamiento de frecuencia. Técnica de transformación de bits en dos frecuencias diferentes que representan 0 y 1 para transmisión en líneas telefónicas o radio. El dispositivo interfaz se llama modem.

Masa. Punto de referencia eléctrica de un circuito.

Copia impresa. Salida impresa en papel.

Hardware. Son los componentes físicos, periféricos u otros equipos que constituyen un sistema de computadora. Contrasta con software.

Hexadecimal. Un sistema de numeración basado en múltiplos de 16 que utiliza los caracteres 0 a 9 y A a F. Por ejemplo, OB hexadecimal es igual a 0000 1011 en binario. Un octeto se puede codificar exactamente en dos símbolos hexadecimales.

Lenguaje de alto nivel. Un lenguaje de programación que es relativamente independiente del lenguaje ensamblador o máquina. La gramática suele parecerse

al inglés y requiere un compilador o interpretador para convertir a código ejecutable. Ejemplos: BASIC, FORTRAN, COBOL, ALGOL, PL/M, APL.

Instrucción. Un paso de un programa que define una operación junto con la/s dirección/es de cualquier dato necesario para la operación.

Interfaz. Es una frontera común entre dos sistemas o dispositivos. Es el hardware o software necesario para interconectar dos partes de un sistema.

Interrupción. Una rotura o interrupción en la ejecución de un programa se suele producir por una señal procedente de un dispositivo externo.

Kansas City estándar. Es un sistema estándar de grabación de datos en cinta de casete que sigue las normas EIA-RS-232C. Ocho ciclos de 2400 Hz iguales a 1, y 4 ciclos de 1200 Hz igual a 0.

Bit menos significativo. Es el dígito binario que ocupa la posición más a la derecha de un número o palabra, esto es: 2.° ó 1.

LIFO. Siglas de Last-in, Firts-out (último en entrar, primero en salir). Método de acceso a la entrada, más reciente, después el siguiente más reciente, y así sucesivamente.

Lápiz óptico. Dispositivo fotosensitivo que se puede utilizar para cambiar la representación en un CRT generando un impulso en el punto de contacto.

Lenguaje máquina. Juegos de enteros binarios que se pueden ejecutar directamente como instrucciones por los microcomputadores sin interpretación previa.

Memoria de masas. Discos flexibles, casetes o cintas utilizados para almacenar grandes cantidades de datos. Son menos accesibles, pero más grandes que la memoria principal.

Memoria. Dispositivo para almacenar información binaria.

Microcomputadora. Un pequeño sistema de computadora capaz de ejecutar un repertorio básico de instrucciones. Incluye una unidad central de proceso, contenida a veces en un único chip, memoria, dispositivos de E/S y fuente de alimentación.

Microprocesador. Una unidad central de proceso en un chip. Un procesador completo en un único chip, fabricado utilizando técnicas de fabricación de microminiatura, conocido como LSI (large scale integration, «integración a gran escala»).

Modem. MOdulador-DEModulador. Dispositivo que transforma datos binarios en frecuencias adecuadas para la transmisión en líneas telefónicas y viceversa.

Monitor. Un programa que controla la operación de rutinas para optimizar el tiempo de computador.

Bit más significativo. El dígito binario que ocupa la posición más a la izquierda en un número o palabra, normalmente 2^7 ó 128.

Octal. Un sistema de numeración basado en múltiplos de ocho utilizando dígitos de 0 a 7. En la actualidad está sustituido ampliamente por el sistema hexadecimal.

Sistema operativo. Software que opera con los recursos software de una microcomputadora. El sistema operativo puede hacer previsión de tiempos, depuración, control E/S, contabilidad, compilación, asignación de memoria y gestión de datos.

Paridad. Un bit adicional que indica si una palabra de computadora tiene un número par o impar de dígitos «1». Utilizado para detectar errores.

Periférico. Cualquier parte del equipo, usualmente un dispositivo de E/S, conectado a la unidad central de proceso.

Memoria programable. Memoria en la que el acceso de nueva información es independiente de la dirección examinada anteriormente.

Memoria de sólo lectura (ROM). Memoria cuyo contenido no se puede modificar. La información se escribe en el momento de su fabricación.

Registro. Un dispositivo de memoria accesible directamente por la unidad central de proceso, utilizado para el almacenamiento temporal de una palabra de computadora durante las operaciones aritméticas, lógicas o de entrada/salida.

S-100. Un bus de 100 terminales (patillas o pines), utilizado en el popular sistema 8080/Z80.

Software. Programas que traducen (convierten) lenguajes de alto nivel en lenguajes máquina, tales como compiladores, sistemas operativos, ensambladores, generadores, librerías de rutinas y editores.

Pila. Una técnica de presentación de programas secuencialmente. Una pila (stack) es una estructura LIFO controlada por instrucciones PUSH (introducción) y POP (extracción).

Mini BASIC. El lenguaje de programación BASIC reducido a una forma simple que permite operaciones aritméticas con enteros y algunas de cadena. Suelen ocupar 4 koctetos o menos de memoria.

Triestado. Capacidad de existencia de tres estados lógicos-o (bajo), 1 (alto), no definido (alta impedancia), es decir, flotante.

UART. Transmisor receptor asíncrono universal. Un transmisor que convierte datos serie a paralelo y *viceversa*.

Palabra. Un conjunto de bits que ocupan una posición de almacenamiento y se trata como una unidad. Puede tener cualquier número de bits, usualmente 4, 8 ó 16.

Procesador de palabras. Un editor de texto que permite al usuario modificar textos: formatos, libros, cartas e informes.

INDICE

Abanico de salida (Fanout), 103
ABC, 57, 68
Acarreo (arrastre), 30
  indicador, 53, 83
Acceso directo a memoria (DMA), 104, 138
Acumuladores, 27, 34
ADC, 53, 65
ADD, 50, 65
Analizadores lógicos, 85, 97, 103
AND, 35, 56
Amplificación intermedia o separación (buffering), 102
  bus de datos, 104
  bus de direcciones, 103
Aplicaciones de la unidad central de proceso, Z80 (ZAP), vii, 1, 19
  comprobaciones, 130, 134
ASCII, 137, 139, 141, 147, 230

BASIC, 139, 193
Bifurcación
  condicional, 83
  incondicional, 82
BIT, 78
Bits:
  comienzo y parada, 148
  indicador, 33
  manipulación, 32, 78
  más significativo (MSB), 194
  menos significativo (LSB), 194
Buses, 22
  alimentación, 102
  amplificación intermedia (buffering), 102
  arquitectura, 25
  control, 104
    comprobaciones, 108
    señales, 106
  controladores, 98, 103, 104
    verificación, 110
  datos, 22, 30, 88, 104, 122
    bidireccional, 23, 104, 110
  de direcciones, 29, 88, 102, 110, 115
  estructuras, 22
    tensión, 19

CALL, 85, 162
Capacitancia, 15
Caracteres, 225
  formato, 227
Cargas, 7, 103
  TTL, 98
Casetes, 127, 137, 155
  interfaz, 120, 155, 158-159
  Software, 158
  Kansas City Standard, 156
CCF, 62
Cero, 28
  indicador, 78, 83
Ciclos de máquina, 30, 96
Circuitos:
  complejidad, 21, 23
  esquemas, 15
  integrados, 10, 22
  protección, 10
  reinicialización (reset), 100
Código de operación, 30
Condensadores, 2, 5-6, 101-102
  constantes de tiempo, 6
  entrada, 15
  filtro, 2, 5, 15
    factor de rizado, 5
  de paso, 15
  tamaño, 5-6
  tiempo de carga, 5
Consideraciones térmicas, 16
Controladores
  bus, 98
  LED, 98
  visualizador, 98
Controladores, inteligente, 193
Convertidores:
  analógico-digital, 194, 199
    CA/CC $3\frac{1}{2}$ dígitos, 211
    analógico a anchura de impulsos, 200
    por aproximaciones sucesivas, 205
    contador binario en rampa, 202
    software, 216
  digital-analógico, 194
    calibración, 198
    multiplicador, 196
    R-2R, 195
    de resistencias ponderadas, 195
COM 2017, 233
COM 8046, 233
Comprobaciones:
  dinámica, 134
  estática, 130
Comunicaciones, 147
  asíncrona, 148, 151
  estándar, 153
  niveles de señal, 153
  paralelo y serie, 137
    software, 158
Corrientes:
  continua, 7
  regulador, 5
  sobrecarga, 6
Cortocircuitos, 19
Coste, 23
CP, 59
CPD, 49
CPDR, 50
CPI, 48
CPIR, 49
CPL, 62
Criterio de Nyquist, 208
CRT 5027, 225
CRT 8002, 225

DAA, 63
Datos, 22, 34, 117, 122
  adquisición, 209, 220
  ASCII, 147
  comunicación, 147
  formatos, 32

frecuencias, 153, 158, 233
orden alto y bajo, 34
DEC, 61, 67-68
Decimal codificado en binario (BCD), 32, 63, 194
Decodificación:
comprobación, 115-116
E/S, 95, 110-111, 114
hexadecimal, 142
memoria, 95, 110-111, 115
Demultiplexores, 114, 218
Desbordamiento, 28
DJNZ, 85
DI, 64
Diodos, 3, 6-7, 102
puentes, 6-7, 17-18
silicio, 3
zener, 8, 10
Diodos emisores de luz (LED), 98, 127
excitadores, 98
Direccionamiento, 29, 32-33, 102, 110
capacidad, 32
orden inferior, 32
orden superior, 32
Disipación de potencia, 4, 16
2114, 125
2102A, 120, 183
2716, 120, 184

EI, 64
Entrada, 22, 88, 129-130
filtros, 2
Entrada/salida, 127, 137
comprobaciones, 130, 134
decodificación, 95, 110
comprobación, 116
escritura, 111
instrucciones, 32, 88
lectura, 111
petición, 31, 111
puertos (ports), 102, 110, 114
registros, 95
Esperas, 31, 96
EX, 45
EXX, 45
Factor de rizado, 5
Faradios, 5
Flip-Flops, 96, 140
Frecuencias de muestreo, 205, 208
Fuentes de alimentación, 1, 16
CC, 1
Fusibles, 18

HALT, 30, 64
HP7340, 144

IN, 88, 129
INC, 60, 67
IND, 90
Indicadores, 33
arrastre, 54, 83
cero (Z), 78, 83
condición, 34, 35
estado, 34
INDR, 90
Inductancia, 14
INI, 89
INIR, 89
Instrucciones, 21
aritméticas y lógicas, 32
de 8 bits, 50
de 16 bits, 65
de uso general, 62
de búsqueda y transferencia de bloques, 32, 45
cambio, 28, 32, 45
carga, 32
8 bits, 35
16 bits, 40
ciclo, 95
ciclo de búsqueda, 29, 95-96
comprobación, 108
conjunto de un solo paso, 96
control de la CPU, 32, 62
ejecución, 96
entrada y salida, 32, 88, 91, 130
extraer, 44
formatos, 32
introducir, 44
llamada y retorno, 32, 85, 162
manipulación de bits, 32, 78
reinicio, 162
rotación y desplazamiento, 32, 68
salto, 32, 82
tipos, 32
Interfases:
CA/CC de $3\frac{1}{2}$ dígitos, 211
comprobación, 217
casete, 155
sintonización, 159
reloj, 221
RS-232C, 225
Serie, 137, 147, 151
Interrupciones, 31, 64, 87
dirección de página, 29
no enmascarables, 31, 87
IM, 64

JP, 82
JR, 82

Kansas City Standard, 156
KR2376, 233

LD, 35
LDD, 47
LDDR, 48
LDI, 47
LDIR, 47
Lógica transistor-transistor (TTL), 97, 102, 230
baja disipación Schottky (LSTTL), 103, 230
cargas, 98
niveles, 151
salidas, 147, 156
Lógica TTL Schottky de baja disipación (LSTTL), 103

Masas, 16
buses, 16
comunes, 15

punto único, 16
referencias, 12
Memoria, 22, 32, 95, 116
de acceso aleatorio (RAM), 122
acceso directo a memoria (DMA), 104
almacenamiento, 117, 127, 155
bancos, 115, 124
comprobaciones, 134
contenidos, 35
decodificación, 95, 110, 115
comprobaciones, 115-116
dinámica, 122
direcciones, 32, 100, 115
escritura, 30, 96, 112
ciclos, 123
estática, 122
lectura, 30, 96, 111
ciclos, 123
lectura/escritura (RWM), 118, 122
lento, 96
mapa, 123
memoria de sólo lectura programable y borrable (EPROM), 118, 121, 162
borradores, 189
programadores, 183
automático, 187
manual, 183
página, 226
petición, 30, 122
posiciones, 28
presentación y sustitución, 161, 163
programable, 27, 115
regeneración («refresco»), 29-30, 122
de sólo lectura (ROM), 115, 117, 183
generador de caracteres, 225
matriz de diodos, 119
programable (PROM), 118
Microcomputadoras, 21
construcción, vii, 27, 95
definición, 21
diseño, 21, 27
sistema, 22
en una sola tarjeta, 193
Microprocesadores (*Véase también* Unidades centrales de tratamiento), 21
arquitectura, 21, 27
corrientes, 24
definición, 21, 27
Z80, 24, 27
Modulación por desplazamiento de frecuencia (FSK), 156
Monitores (*Véase también* Software), 120, 125, 141, 161, 183
arranque en caliente, 161, 162, 170
arranque en frío, 161
diagnóstico UART, 166
ejecutar, 161, 165, 182
entrada/salida serie, 161, 166-167, 168-169
entrada de teclado, 174
reconocimiento de órdenes, 172
reinicio, 173
visualización de memoria y sustitución, 161, 163, 179
visualización de registro y sustitución, 161, 164, 179
Multiplexores, 23, 123

NEG, 62
Ninguna operación (NOP), 31, 32, 63-64

Octetos (bytes), 32
8080A, 24, 32, 95
8212, 105
Ondas sinusoidales, 2
Operandos, 35-36
OR, 35, 57
Osciloscopios, 95, 97
OTDR, 93
OTIR, 92
OUT, 91, 128
OUTD, 92
OUTI, 91

Paridad, 28
Pascal, 193
Periféricos, 127, 137, 161
sincronización, 138
Peticiones, 111
entrada/salida, 111
escritura, 111
lectura, 111
memoria, 111
Pilas, 28, 32, 43, 85, 112
POP, 44
Programas:
depuración, 163
desarrollo, 163
Protectores de sobretensiones, 18
Puentes rectificadores de onda completa (*Véase también* Rectificadores), 2, 6
Puertos (Ports), 34, 88, 110,114
octal, 144
paralelo y serie, 138, 193
salida hexadecimal, 144
PUSH, 44

Radiador de calor, 17
Rectificadores (*Véase también* puentes de onda completa), 6, 15
controlado por silicio (SCR), 19-20
puente, 2, 6, 17
onda completa, 3, 6
Refrigeración, 18
Regeneración («Refresco»), 29-30, 122
Registros, 27-28
acumulador (A), 27-28, 34
8 bits (B, C, D, E, H, L), 27, 117
16 bits (BC, DE, HL), 27
conjuntos, 27-28
contador de programa (PC), 28, 33, 82, 85, 162
contenido, 35
de dirección de interrupción de página (I), 29
indicador (F), 27-28, 33
índice (IX, IY), 29
instrucción, 29
pares, 28, 33, 40
principal y alternativo, 28-29
presentación y sustitución, 161, 164
punteros de pila (SP), 28, 43, 162
refresco de memoria (R), 29
de uso general, 28
Reguladores de tensión (*Véase también* Tensiones, reguladores)
Reinicializar (Puesta a cero, Reset), 64, 100, 162

automático, 101
comprobaciones, 108-110, 134
manual, 101
Relojes, 95, 221
comprobación, 108-110
de paso único, 97, 110
períodos, 96
tiempo real, 220
RES, 81
Resistencia, 4, 6, 16
serie, 6, 8
térmica, 17
Resistencias (componentes), 19, 195
en escalera, 195
variable, 8
Resolución, 194, 197-198, 209
RET, 86
RETI, 87
RETN, 87
RL, 71
RLA, 68
RLC, 69
RLCA, 68
RLD, 77
RR, 73
RRA, 69
RRC, 72
RRCA, 69
RRD, 78
RS-232C, 153, 225
RST, 87, 162

Salida, 22, 91, 128
SCF, 63
Sección de control, 22
6502, 24
6800, 24
Selección de chip, 122
SCF, 63
Signo, 28
Sistemas operativos, 161
78H05, 11, 17
7812, 12
7912, 12
SLA, 74
Sobrecargas, 6
Software (*Véase también* Monitores), 24
escalonamiento, 97
monitores, 161
SRA, 75
SRL, 76
Strobe (Habilitación
datos listos, 138
duración, 140
tecla-oprimida, 148
SUB, 54
Subrutinas, 28, 85, 125

Teclados, 120, 127, 137
ASCII, 137, 141
codificadores, 140-141, 233
hexadecimal, 141
rebote, 140
software de entrada, 174
Tarjetas de circuito impreso, 21
Temporizadores, 138
Tensiones:
caídas, 3, 7, 11, 15
cargas, 5
convertidores, 7-9
comparadores, 7-9
corriente alterna, 2
corriente continua (DC), 1
elemento de control, 8
entradas y salidas, 7, 15
formas de onda, 2-3
inversa de pico (PIV), 6
ondas senoidales, 2
pico, 4, 17
referencia, 8, 10
reguladores, 1, 2-3, 7, 11, 17
elección, 11
series, 8
sobrecargas, 10
tres terminales, 10
rizado, 4-5, 15
transistores, 6
VCA, 2
valor eficaz (RMS), 2, 7
Tensiones inversas de pico (PIV), 6
Terminales, 226
Transformadores, 2, 7
entrada a primario, 2
salida de secundario, 2-3
Transistores, 8, 18
banda ancha, 15
conducción en serie, 10
FAMOS, 121, 183
Transmisor receptor asíncrono universal (UART), 148, 233
diagnóstico, 166
diagrama de conexiones, 148
salida, 155

Unidad aritmética y lógica (ALU), 22, 29
Unidades centrales de proceso (*Véase también* Microprocesadores), 21-22, 27
arquitectura, 27
control, 29, 32
comprobación, 134
estados, 33
registros, 27, 29
sincronización, 102
temporización, 96

Visualizadores:
diodo emisor de luz (LED), 98, 127, 137, 142, 163
hexadecimal, 142
octal, 142
tubo de rayos catódicos (TRC), 137, 147, 225
vídeo, 127, 193, 225
visual, 85
Voltímetro, 98, 194, 211

XOR, 58

Z80, 24, 27
diagrama de conexiones, 29
estructura de bus, 25
ocupación lógica de control, 95

NOTAS

NOTAS

NOTAS

NOTAS

NOTAS

NOTAS

NOTAS

NOTAS

OTRAS OBRAS DE INTERES
PUBLICADAS POR OSBORNE/McGRAW-HILL

ADAM OSBORNE: Guía del comprador de sistemas de gestión
ANNIE FOX: BASIC Básico (Guía para principiantes)
ADAM OSBORNE: Guía del ordenador personal PET/CBM
JOHN HEILBORN: Programas para ciencias e ingeniería. Edición APPLE II
JOHN HEILBORN: VIC 20. Guía del usuario
LON POOLE: Algunos programas de uso común en BASIC
LON POOLE: APPLE II. Guía del usuario
LON POOLE: Algunos programas de uso común en PASCAL
LON POOLE: Programas prácticos en BASIC. Edición APPLE II
LON POOLE: Programas prácticos en PASCAL
LON POOLE: Algunos programas de uso común en BASIC. Edición IBM
LON POOLE: Algunos programas de uso común en BASIC. Edición PET/CBM
LON POOLE: Algunos programas de uso común en BASIC. Edición TRS-80
LON POOLE: Algunos programas de uso común en BASIC. Edición APPLE II
LON POOLE: Algunos programas de uso común en BASIC. Edición ATARI
LON POOLE: Programas prácticos en BASIC.
LON POOLE: Programas prácticos en BASIC. Edición TRS-80
LON POOLE: Programas prácticos en BASIC. Edición IBM
ROBERT MOTTOLA: Programación en lenguaje ensamblador para el APPLE II
TOM HOGAN: Sistema operativo CP/M. Guía del usuario. 2.ª ed.
WALTER ETTLIN: Introducción al WORDSTAR. 2.ª ed.

OTRAS OBRAS DE INTERES
PUBLICADAS POR BYTE-BOOKS/McGRAW-HILL

MORGAN: Introducción al micro procesador 8086/8088 (16 bit)
KAMINS: Usted y el «APPLE» (Una introducción humanizada a la microinformática)

ISBN: 968-451-564-2